INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 16, No. 5
October 2018
ISSN: 1545-679X

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **ISCAP** (Information Systems and Computing Academic Professionals). Publishing frequency is six times per year. The first year of publication was 2003.

ISEDJ is published online (http://isedj.org). Our sister publication, the Proceedings of EDSIGCON (http://www.edsigcon.org) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the EDSIGCON conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to members of AITP-EDSIG who perform the editorial and review processes for ISEDJ.

INFORMATION SYSTEMS EDUCATION JOURNAL

Information and Communication Technology in the Classroom: BYOD and the University's Role

Gary Alan Davis
davis@rmu.edu

Frederick G. Kohun
kohun@rmu.edu

Computer and Information Systems
Robert Morris University
Moon Township, PA 15108, USA

Abstract

Personally-owned laptops and other *Bring-Your-Own-Device* (BYOD) scenarios have become increasingly prevalent in today's work environments and classrooms. However, few studies have examined the viability and practicality of such devices in the higher-education classroom. This study used a survey instrument to explore the concept of BYOD in the classroom. Specifically, undergraduate and graduate students were asked to report their use (both inside and outside of the classroom) of personally-owned devices, and their use of university-managed computer labs. The findings of this research will be of interest to higher-education faculty, administrators, and Information Technology departments.

Keywords: Laptop, Information and Communication Technology (ICT), Bring-Your-Own-Device (BYOD), Higher education

1. INTRODUCTION

The use of personal computing devices is becoming increasingly popular within business and within other professional organizations. A 2012 survey by Cisco of U.S. organizations found that 95% of the respondents allowed some form of *Bring-Your-Own-Device* (BYOD) in the workplace (Kaneshige, 2012). In fact, BYOD has been frequently described as the " . . . most radical shift in the economics of client computing for business since personal computers invaded the workplace" (Willis, 2012, p. 1).

Laptop computers have been referred to as the most used and most important devices for academia (Dahlstrom, Walker, & Dziuban, 2013). Although personal laptops have long been used in higher-education, their efficacy in the learning process has been fiercely debated. On one side of the debate, the use of laptops in the classroom has been found to keep students on task, increase students' capabilities for following lectures, and foster collaboration among students (Kay & Lauricella, 2016). On the other side of the debate, some researchers have noted many "off-task" uses of laptops by students, such as surfing social media, playing games, and watching videos and movies (Barak et al., 2006; Barkhuus, 2005).

Extensive research has also been conducted in the use of laptops in primary and secondary education (i.e., "K-12" grades). However, the implementation of laptops in K-12 education has been predominantly limited to *1:1 laptop initiatives*. In 1:1 laptop initiatives, the school district provides each student with his/her own laptop for classroom use (Tallvid, Lundin, Svensson, & Lindstrom, 2015).

Although the educational effectiveness of personal computing devices (e.g., laptops) has been widely researched, few authors have examined higher education's role in providing (or requiring) such devices. Specifically, what is the university's role in providing (or requiring) *Information and Communication Technology* (ICT) in the modern, higher-education classroom?

The purpose of this study was to survey undergraduate and graduate students to determine their use of personally-owned computing devices (e.g., laptops), and their use of university-provided computing devices (i.e., computer labs). Specifically, the study sought to answer the following research questions:

1. What percentage of students have their own personal laptop computer?
2. What percentage of the time do students use *personally-owned* laptops for homework and/or lab assignments both *within-class* and *outside-of-class?*
3. What percentage of the time do students use *university-provided* labs for homework and/or lab assignments both *within-class* and *outside of class?*
4. Is there a statistically significant difference in the use of *personally-owned* laptops and *university-provided* labs (for both *within-class* and *outside-of-class* work)?

2. DEFINITION OF TERMS

Information and Communication Technology (ICT) is defined as " . . . integrated systems which are capable of handling and linking up many types of information: written and spoken languages, still and moving visual images, and data of all kinds" (Adeyoyin, Okunlaya, Alawiye, & Emmanuel, 2013, p. 191). *Bring-Your-Own Device* or BYOD, are corporate policies that " . . . encourage practices of allowing employees to use their personally owned mobile devices to conduct their work, whether inside or outside of their workplaces" (Garba, Armarego, Murray, & Kenworthy, 2015, p. 38). In this definition, "mobile devices" could refer to laptop computers, or any other mobile computing device, such as a tablet or smartphone. Finally, *1 to 1 (i.e., 1:1) laptop initiatives* are programs in K-12 schools, where each student receives a laptop, from the school district, to " . . . supplement their regular classroom learning" (Hatakka, Andersson, & Gronlund, 2012, p. 94).

3. REVIEW OF LITERATURE

Current research studies into ICT and BYOD fall into three main categories: 1) educational effectiveness of 1:1 laptop initiatives, 2) impacts of BYOD policies in the workplace, and 3) the design and construction of BYOD-friendly environments (both in industry and in academia).

Brousard, Hebert, Welch, and VanMetre (2014) conducted focus groups and classroom observations of 650 students and 40 teachers in order to evaluate a 1:1 laptop initiative at a secondary school. The authors determined that the 1:1 laptop initiative in their study fostered a "flipped" classroom, in which the learning shifted from "teacher-focused" to "student-focused." The authors also found that the use of laptops in the classroom encouraged the teachers to use more "technology-rich" content in their instruction (p. 42).

Tallvid, Lundlin, Svensson, and Lindstrom (2015) collected data from 500 students over a three-year period to determine what uses of a 1:1 laptop initiative were "sanctioned" (i.e., education-related), and what uses where "unsanctioned" (i.e., not education-related). While the authors noted a significant percentage of "unsanctioned" use among students (e.g., playing games or watching movies), the research findings suggested that, as overall laptop use increased, both "unsanctioned" and "sanctioned" use of the laptops increased.

Finally, Tallvid (2016) conducted a qualitative follow-up study of 60 teachers to determine why some teachers were reluctant to adopt ICT as part of a 1:1 laptop initiative. Tallvid discovered "patterns of reluctance" among the teachers, such as "lack of technical competence, not worth the effort, insufficient material, diminishing control, and lack of time" (p. 503).

Overall, the findings from 1:1 laptop initiatives have been mixed. Some researchers have suggested that " . . . a link exists between 1:1 programs and student achievement" (Downes & Bishop, 2015, p. 2). However, other studies have revealed conflicting results. For example, Hur and Oh (2012) found that while 1:1 laptop programs did raise student engagement, there was no statistically significant improvement in students' test scores.

Studies involving BYOD policies in the workplace have primarily focused on information security and privacy risks. Garba, Armarego, Murray, and Kenworthy (2015) examined the benefits and costs associated with BYOD policies. The authors found that cost savings from BYOD can be realized, such as reduced travel, facility, device, and data service costs. However, if not addressed, the risks to data security and privacy can outweigh the benefits of BYOD. The authors suggest that any organization considering BYOD must " . . . strike a balance between the availability and protection of information resources and assets" (p. 51).

Researcher Chris Rose (2013) took the cost benefit analysis of BYOD one step further. Rose not only looked at information security and privacy concerns, but also the branding and legal liability associated with BYOD. The author concluded "BYOD might initially sound like a bargain but the loss of brand identity, the possibility of legal liability, the difficulty of IT departments supporting different phone/version/carrier combinations and the many security problems . . . " may negate any anticipated benefits of BYOD (p. 68).

Finally, there is a growing body of research involving organizations that are designing specific BYOD-friendly spaces. For example, Dallis (2015) developed a case study from a facility redesign at Indiana University at Bloomington. The 27,000-square foot University Library was redesigned specifically to reflect "bring-your-own device interior designs" (p. 47). Even commercial airlines are redesigning their planes to cater to BYOD passengers. American Airlines recently announced that it is eliminating the seat-back screens from its new Boeing 737 Max jets (Ostrower, 2017). The announcement was made after the airline determined that 90% of its passengers bring their own mobile devices onboard. American Airlines states that "smartphones, tablets or laptops do a better job than the airline's individual screens" (p. 1).

4. RESEARCH METHODOLOGY

The current study involved an online survey that was completed by undergraduate and graduate students within *Computer and Information Systems* (CIS) courses at a private, medium-sized university. Participation in the survey was voluntary, and all responses were anonymous. The survey was created using *QuestionPro* survey software, and was available from November 17 to December 6, during the Fall semester of 2016.

The online survey consisted of 22 questions. Most of the questions were closed-ended, however, some questions provided an open-ended field so participants could elaborate on their answer. A total of 322 students opened the survey, 220 students began the survey, and 200 students completed the survey in its entirety. The 200 students who completed the survey make up approximately 26% of the total enrollment for CIS degree majors at the university. It should be noted that only the 200 completed survey responses were used in the current study (i.e., no incomplete surveys were considered in this study).

QuestionPro survey software was used for descriptive statistics and basic data analysis. For statistical testing, the survey responses were imported into IBM *SPSS* (Statistical Package for the Social Sciences) version 24.0. **Table 1: Participant Degree Type** and **Table 2: Participant Degree Program** in **Appendix A** show the demographic breakdown of the survey participants.

5. RESULTS

In order to address the first research question, "What percentage of students have their own personal laptop computer?," the survey participants were asked if they currently have their own laptop computer. Out of the 200 completed surveys, 186 (93.0%) participants reported that they owned a personal laptop computer.

Reviewing the results in terms of degree type, 139 (93.9%) undergraduate students stated that they owned a personal laptop. Twenty-nine (90.6%) Integrated (i.e., 5-year Bachelor's / Master's program) students reported owning a personal laptop. Finally, 18 (90.0%) graduate students reported owning a personal laptop. The results from the first research question are depicted in **Appendix B**, **Table 3: Student Ownership of Personal Laptop Computers.**

Participants who owned a personal laptop were also asked several follow-up questions, such as the age of the laptop and the operating system of the laptop. Fifty-three participants (28.5%) reported that their laptop was one year old or less. Fifty-nine participants (31.7%) reported owning a laptop that was two years old. Finally, 74 participants (22.6%) said they owned a laptop that was three years old or older.

In terms of the operating systems installed on the participants' laptops, the majority (59.1%) of personal laptops were running Microsoft Windows 10. According to participants, 15.1% of the laptops were running Windows 8, and 18.2% were running Apple's OS X. Finally, 7.0% of the participants reported that their laptops were running an operating system described as "Other." The "Other" operating systems reported by participants included the following: Windows Vista, Windows 7, Debian 8, Red Hat Linux, and Linux Ubuntu.

The second research question proposed was, "What percentage of the time do students use *personally-owned* laptops for homework and/or lab assignments both *within-class* and *outside of class?"* To address this question, participants were asked to report both the amount of time that they use their personal laptop for homework or lab work *within-class*, and the amount of time that they use their personal laptop for homework or lab work *outside-of-class*. The following Likert-like scale was used to allow the participants to report the amount of work completed (both inside and outside-of-class) with their personal laptop: 5 = Greater than 75% of work completed, 4 = 75% of work completed, 3 = 50% of work completed, 2 = 30% of work completed, 1 = 20% or less of work completed.

As for *within-class* usage of personal laptops, the mean score reported by participants was 2.48. This score indicates that participants reported using their personal laptops for 30 to 50% of *within-class* work assignments.

In regard to *outside-of-class* usage of personal laptops, the mean score reported by participants was 3.68. This score indicates that participants reported using their personal laptops for 50 to 75% of *outside-of-class* work assignments. The results from the second research question are depicted in **Appendix C**, **Table 4: Student Use of Personal Laptops versus University Lab PCs.**

The third research question proposed was, "What percentage of the time do students use *university-provided* labs for homework and/or lab assignments both *within-class* and *outside of class?"* To address this question, participants were asked to report both the amount of time that they use a university-provided lab computer for homework or lab work *within class*, and the amount of time that they use a university-provided lab computer for homework or lab work *outside-of-class*. Again, the previously described 1 to 5 "usage scale" was used.

As for *within-class* usage of a university lab computers, the mean score reported was 2.22. This score indicates that participants reported using university-provided labs for 30 to 50% of *within-class* work assignments.
Regarding *outside-of-class* usage of university lab computers, the mean score reported by participants was 1.81. This score indicates that participants reported using university-provided labs for 20 to 30% of *outside-of-class* assignments. The results from the third research question are depicted in **Appendix C**, **Table 4: Student Use of Personal Laptops versus University Lab PCs.**

As shown in **Table 4**, the mean usage score of personal laptops reported by students is higher than the mean usage score of university-provided labs for both *within-class* work and *outside-of-class* work. **Table 4** also shows that the difference in mean usage scores is greater for *outside-of-class* work. The difference in mean scores does not, however, reveal if the difference between personal laptop usage and university lab usage is at a level that is statistically-significant.

The fourth and final research question explored whether or not there was a statistically-significant difference in usage between personally-owned laptops and university-provided lab computers. As in research questions two and three, student usage was measured in terms of both *within-class* work assignments and *outside-of-class* work assignments. The *Paired-Samples T-Test* was used to determine if the difference in mean scores was statistically significant at the .05 confidence level.

In analyzing the usage of personal laptops compared to university-provided labs for *outside-of-class* work, there was a significant difference in the mean scores for personal laptop usage (M=3.68, SD=1.312) and university lab usage (M=1.81, SD=1.282); $t(196)=12.852$, $p=.000$. In analyzing *within-class* work, however, there was not a significant difference in the mean scores for personal laptop usage (M=2.48, SD=1.473) and university lab usage (M=2.22, SD=1.410); $t(194)=1.576$, $p=.117$.

To thoroughly address the last research question, the current research also compared the *overall* mean usage scores between personal laptop usage and university-provided lab usage (i.e., regardless of whether the work was performed *within-class* or *outside-of-class*). Overall, there was a significant difference in usage between personal laptops (M=3.08, SD=1.516) and university-provided labs (M=2.02, SD=1.361); $t(391)=9.164$, $p=.000$. The Paired-Samples T-Test results are depicted in **Appendix C, Table 5: Paired Samples T-Test.**

6. DISCUSSION AND CONCLUSIONS

The context for this study was research into whether or not a standalone computer laboratory for information system computing majors as mandated by *ABET-CAC* accreditation was, in fact, a necessary and value-added resource. It appeared to the researchers that, with the student proliferation of BYOD in the classroom, dedicated computer laboratories may not have as crucial a role as was in the past. With a virtual machine environment available to all mobile devices (i.e., VMware, aka Horizon), students have a viable option that has emerged over the past few years. While subject to numerous variables, such as Wi-Fi speed, allocated memory availability in the virtual server, software licensing issues, and configuration setup expertise, virtual machine technology has offered students a robust alternative to standalone computer laboratories.

The survey results from the current study have shown that 93.9% of undergraduate students own a laptop computer, however, when excluding tablets and other mobile devices, only 4.5% of the surveyed undergraduate population did not own a laptop computer. With respect to the integrated undergraduate/graduate students (5-year Bachelor's/Master's degree) 90.6% owned laptop computers with 1.5% not owning them (the gap again explained by tablets and other mobile devices). Additionally, the survey of graduate students indicated that 90% owned laptop computers.

The findings from the survey related to computer laboratory usage and BYOD usage. In the context of classroom use of computers, the survey yielded a virtual split between students using University lab computers (mean of 2.22) and personal laptops (mean of 2.48). However, with respect to outside of class usage of computers to do assigned work, a resounding majority (mean of 3.68) used their personal laptops with a significantly smaller number (mean of 1.81) using university computer in the laboratory. Using a t-test, the survey demonstrated statistical significance (p=.000) in the difference between personal computer usage and university laboratory computer usage for out of class assignments.

It can be concluded that with the convenience, lower cost point, cultural affinity toward mobile computing, and efficient and cost effective virtual machine (i.e., cloud) availability of specialized software, students tend to prefer BYOD rather than utilize a dedicated university computer laboratory. As virtual machine capabilities improve, as specialized software is adapted to cloud environments, and as Wi-Fi security and reliability improves, we can see further increased use of BYOD mobile devices with less use and value-added associated with dedicated software-focused teaching labs. This questions the need for extensive dedicated computer laboratories for teaching purposes.

The above findings do not imply that special purpose computer responses should not be available for out-of-class work or special research projects. What it does question is the need for universities to allocate significant computer technology resources for teaching classrooms. Universities and accreditation groups, such as ABET-CAC, should consider furthering the discussion on virtual machine technologies, accreditation-required dedicated open labs, and required student laptop ownership. Student computer usage patterns for both classroom and laboratories have changed and continue to change. These changes have had a significant impact on curriculum, overall teaching and learning effectiveness, and accreditation, as well as efficient and effective financial resource allocation.

7. REFERENCES

Adeyoyin, S., Okunlaya, R., Alawiye, M., & Emmanuel, S. (2013). Information communication technology (ict) and national development: librarians' perspective. *Library Progress International,* 33(1), 189-201.

Armando, A., Costa, G., Merlo, A., & Verderame, L. (2015). Formal modeling and automatic enforcement of bring your own device policies. International *Journal of Information Security,* 14(2), 123-140.

Barak, M., Lipson, A., & Lerman, S. (2006). Wireless laptops as means for promoting active learning in large lecture halls. *Journal of Research on Technology in Education*, 38(3), 245-263.

Barkhuus, L. (2005). Bring your own laptop unless you want to follow the lecture: Alternative communications in the classroom. In *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work* (pp. 140-143), New York, ACM.

Broussard, J., Hebert, D., Welch, B., & VanMetre, S. (2014). Teaching today for tomorrow: A case study of one high school's 1:1 computer adoption. *Delta Kappa Gamma Bulletin, 80*(4), 37-45.

Dahlstrom, E., Walker, J. D., & Dziuban, C. M. (2013). *The ECAR study of undergraduate students and information technology (Research Report)*. Louisville, CO: EDUCAUSE Center for Analysis and Research.

Dallis, D. (2016). Scholars and learners: A case study of new library spaces at indiana university. *New Library World, 117*(1), 35-48.

Downes, J. M., & Bishop, P. A. (2015). The intersection between 1:1 laptop implementation and the characteristics of effective middle level schools. *RMLE Online, 38*(7), 1-16.

Garba, A. B., Armarego, J., Murray, D., & Kenworthy, W. (2015). Review of the information security and privacy challenges in bring your own device (BYOD) environments. *Journal of Information Privacy & Security, 11*(1), 38-54.

Hatakka, M., Andersson, A., & Grönlund, Å. (2013). Students' use of one to one laptops: A capability approach analysis. *Information Technology & People, 26*(1), 94-112.

Hur, J. W. & Oh, J. (2012). Learning, engagement, and technology: Middle school students' three-year experience in pervasive technology environments in South Korea. *Journal of Educational Computing Research*, 46(3),295-312.

Kaneshige, T. (2012). CIO.com CIO Challenge with BYOD: Don't fall down the rabbit hole. Retrieved August 24, 2012 from http://www.cio.com/article/706579.

Kay, R., & Lauricella, S. (2016). Assessing laptop use in higher education: The laptop use scale. *Journal of Computing in Higher Education, 28*(1), 18-44.

Ostrower, J. (January 25, 2017). American airlines eliminating in-seat screen on new jets. Retrieved from http://money.cnn.com/2017/01/25/technology/american-dropping-screens-from-boeing-737-max/.

Rose, C. (2013). BYOD: An examination of bring your own device in business. *The Review of Business Information Systems (Online), 17*(2), 65.

Tallvid, M. (2016). Understanding teachers' reluctance to the pedagogical use of ICT in the 1:1 classroom. *Education and Information Technologies, 21*(3), 503-519.

Tallvid, M., Lundin, J., Svensson, L., & Lindström, B. (2015). Exploring the relationship between sanctioned and unsanctioned laptop use in a 1:1 classroom. *Journal of Educational Technology & Society, 18*(1), 237-249.

Willis, D.A. (2012). Gartner publishes 'Bring your own device: New opportunities, new challenges' report. *Entertainment Close-Up.*

Appendix A

Table 1: Participant Degree Type

Type of Degree	Frequency	Percent	Cumulative Percent
Undergraduate	148	74.0	74.0
Integrated[1]	32	16.0	90.0
Graduate	20	10.0	100.0
Doctoral	0	0.0	100.0
Total	**200**	**100.0**	**100.0**

[1] - The *Integrated* program is a 5-year, combined Bachelor's/Master's program

Table 2: Participant Degree Program

Type of Degree	Degree Program	Frequency	Percent	Cumulative Percent
Undergraduate (B.S.)	Computer and Information Systems	65	28.0	28.0
	Cyber Forensics and Information Security	58	25.0	53.0
	Data Analytics	5	2.2	55.2
	Information Science	4	1.7	56.9
	Other[2]	48	20.7	77.6
Graduate (M.S.)	Data Analytics	23	9.9	87.5
	Cyber Security and Information Assurance	15	6.5	94.0
	Information Systems Management	4	1.7	95.7
	Internet Information System	3	1.3	97.0
	Engineering	7	3.0	100.0
Total		**232**[3]	**100.0**	**100.0**

[2] - The category *Other* predominantly included *Engineering*, *Accounting*, and *Actuarial Science*
[3] - 32 Student participants are counted twice due to the *Integrated Bachelor's/Master's Degree* program

Appendix B

Table 3: Student Ownership of Personal Laptop Computers

Type of Degree	n	Own Laptop Frequency	Own Laptop Percent
Undergraduate	148	139	93.9
Integrated	32	29	90.6
Graduate	20	18	90.0
All Degree Types	**200**	**186**	**93.0**

Appendix C

Table 4: Student Use of Personal Laptops versus University Lab PCs

ICT Use by Students		n	Mean	Std. Dev.	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Min.	Max.
Within-Class	Personal Laptops	195	2.48	1.473	0.105	2.375	2.585	1.0	5.0
	University Lab PCs	200	2.22	1.410	0.101	2.119	2.321	1.0	5.0
Outside-of-Class	Personal Laptops	197	3.68	1.312	0.093	3.587	3.773	1.0	5.0
	University Lab PCs	200	1.81	1.282	0.091	1.719	1.901	1.0	5.0

Table 5: Paired Samples T-Test

	Personal Laptops		University Lab PCs		Difference		
Factor	Mean1	Std. Dev.1	Mean2	Std. Dev.2	Mean Diff.	t-Stat.	Sig.
Within-Class	2.48	1.473	2.22	1.410	0.26	1.576	.117
Outside-of-Class	3.68	1.312	1.81	1.282	1.87	12.852	.000
Combined	3.08	1.516	2.02	1.361	1.06	9.164	.000

Cyber Security Curriculum Development: Protecting Students and Institutions While Providing Hands-On Experience

Jim Marquardson
jimarqua@nmu.edu
College of Business
Northern Michigan University
Marquette, MI 49855, USA

David L. Gomillion
dgomillion@mays.tamu.edu
Mays Business School
Texas A&M University
College Station, TX 44843, USA

Abstract

Demand for graduates with cybersecurity skills continues to increase. Many universities have developed or are in the process of developing new courses or degree programs to meet student demand and fill the skill gap in industry. Instructors face unique challenges when developing cybersecurity courses: While it is widely recognized that hands-on exercises are critical for helping students reach course learning objectives, legal, operational, and pedagogical challenges make it difficult to create safe, secure, and reusable exercises. The purpose of this article is to provide course designers principles for developing cybersecurity exercises in a way that maximizes student success while minimizing organizational risk. A matrix to help educators and administrators evaluate controls is provided, allowing for a clearer description of the risks eliminated, mitigated, and accepted. The principles provided in this article are based on the experience of developing a new cybersecurity degree program at a Midwestern university.

Keywords: Cybersecurity, Information assurance, Curriculum design and development, Computer Security, Risk assessment

1. INTRODUCTION

Cybersecurity challenges pervade the global computing infrastructure. Tens of thousands of data breaches incur millions in losses annually across the globe and across industries (Verizon, 2016). Democratic elections face hacking threats (Fidler, 2016). Fears regarding critical infrastructure have prompted governments to invest heavily in cybersecurity (Wagner, 2016). Cybersecurity concerns show no sign of abating. In such an environment, it is unsurprising that there is an increased demand for qualified cybersecurity professionals (Bernstein, 2013).

Institutions of higher education have moved quickly to create degree programs and classes to prepare students for cybersecurity careers. A common challenge that educators face is creating effective and realistic course exercises to teach cybersecurity skills without putting their institutions at significant risk. Hands-on, experiential learning has been shown to be effective in learning about information systems (Jewer & Evermann, 2015). Practicing professionals have stated that, “[t]here must be a strong emphasis on practical exposure to concepts in terms of hands-on experience for students” (Sauls & Gudigantala, 2013, p. 72).

Cybersecurity exercises, by their nature, present significant risk of causing harm, something unique to cybersecurity.

At the risk of stating the obvious, all cybersecurity exercises should be conducted legally. Passive defensive measures like firewalls, anti-virus, and updating operating systems are safe activities that stand little chance of offending the law. However, students need to learn active defensive tools and offensive hacker tools and techniques to know how to protect systems. Students must only perform these security exercises on systems with explicit authorization.

The current state of the art is to perform testing in an isolated environment, typically using virtualization and a segmented network. This is an effective step in protecting the organization, but it may not be enough. Virtualized labs suffer from limitations in scope and experience. Simulating a connection to a social media platform can pose significant challenges to instructors because of the complexity of systems behind those platforms. In addition, recognizing real threats from outside actors requires experiencing these attacks. This leads many instructors to transition away from the isolated lab towards live security testing, which can touch real networks and the internet.

With live security testing, protecting your institution's network and reputation can be difficult for two major reasons: inadvertent student mistakes and purposeful malicious use of their newly developed skills (Nurse et al., 2014). An example of an accidental breach could be a port scanning exercise. Network administrators scan open ports on network hosts to assess their systems, but hackers also scan ports to probe potential victims for exploitable weaknesses. While most security experts may not consider port scanning to be malicious, prosecution is a very real possibility. For example, Scott Moulton scanned a Cherokee County, Georgia web server and was charged with violating the Computer Fraud and Abuse Act of America (Lyon, 2008). Though Moulton was eventually acquitted, the case shows that even seemingly innocuous activities can be interpreted as malicious crimes. A classroom port scanning exercise might ask students to scan a specific internet protocol (IP) address. With an active internet connection, a mistyped IP address could cause a port scan to be conducted on a computer for which the student does not have the required authorization. It is imperative that safeguards are put in place to ensure that simple mistakes do not land students or educational institutions in legal trouble. Universities also must protect their reputation by not engaging in activity that appears to be illegal.

Malicious insiders are individuals within an organization who intentionally abuse acceptable use policies and intend to do the organization harm (Cappelli, Moore, & Trzeciak, 2012). It is essential to teach students the tools and techniques used by malicious actors; unfortunately, a portion of students may employ those tools and techniques in unauthorized ways. For example, one student used keylogging software to steal credentials that allowed him to alter his grades (Osborne, 2012). In another instance in which the present authors were made aware, a member of a student cybersecurity club learned about session hijacking and used that knowledge to view a peer's private Facebook data without permission. It is important to note that these examples occurred outside of formal classroom exercises but leveraged skills taught in the cybersecurity programs. Identity theft, Federal Educational Rights and Privacy Act (FERPA) violations, denial of service attacks, and cyberbullying are just some of the examples of what malicious actors inside campus environments might carry out. Actions taken by network administrators to protect network perimeters are often ineffective against tools and techniques initiated by insiders (Harrison, 2005).

But how can educators anticipate all the risks inherent in cybersecurity curriculum? And how can they communicate the protections put into place so that risk managers at institutions of higher learning can be comfortable with level of risk the organization is accepting? In the following sections we discuss pedagogical considerations, the risk assessment process, controls, and actions that educators can take to protect their students and their organizations. Educators can use the guidance in this paper to ensure that their universities are employing sufficient resources and attention to keeping their organizations safe while delivering value to students.

2. PEDAGOGY

Universities have chosen to create cybersecurity programs from a variety of perspectives. Cybersecurity programs are currently housed in business, computer science, computer engineering, criminal justice and other departments. Depending on the program, emphasis may be given to network administration, penetration testing, forensics, legal matters, to name just a few key areas of cybersecurity. These different perspectives provide students with diverse educational backgrounds to provide complementary skills to the workplace. However, the distinct curriculum at institutions makes it difficult to provide one all-encompassing guide for delivering a cybersecurity curriculum. This extends to ensuring the curriculum is taught safely. Therefore, the first step to ensuring that cybersecurity exercises are done safely it to determine the scope of the curriculum.

Once the learning objectives have been identified, activities to meet those objectives must be planned. In research, tension exists between research methods. Research methods can excel in rigor, relevance, or generalizability, but not all three simultaneously (McGrath, Martin, & Kulka, 1982). A similar challenge exists in pedagogy. Internship experience is highly relevant, but those experiences may not be generalizable across industries, and due to business needs, the work may lack educational rigor. A virtual lab environment can be tuned to provide educational rigor at the expense of relevance. Instructors can focus teaching on principles that can be broadly applied in many contexts, but in so doing usually must relax rigor and relevance.

The process of developing effective and clear objectives is an important topic but beyond the scope of the current work. But once those objectives are defined, the risks of exercises used to reach those objectives must be weighed against the educational value. We provide a risk assessment process to help in identifying controls to help address the identified risks.

3. RISK ASSESSMENT

Risk assessments are key activities undertaken when information systems are deployed (Dhillon & Torkzadeh, 2006). Educators should practice what they preach and perform risk assessments on their own cybersecurity exercises. When assessing risk, educators must thoroughly assess the different ways in which an exercise could cause harm. The primary goals of risk assessments in the cybersecurity exercise context are to ensure that exercises are performed legally and prevent harm to infrastructure. Depending on the risks identified, the risk assessment may not need formal documentation and organizational sign-off, but risks should be evaluated and proper controls should be put in place. But in other exercises, the proper organizational authorizations must be obtained.

Controls must be put in place to protect institutions from both the accident prone and the malicious insider based on the risks identified. Controls such as using virtual labs or isolated network segments can prevent some accidental or malicious behaviors. But it is important to assess risk beyond the classroom for two major reasons: first, live security testing is often required to give students the knowledge, skills, and abilities they will need to be successful; and second, the knowledge and skills learned in the class can be applied outside of the classroom. While it is generally impossible to eliminate risk, based on the risk assessment, different types of controls may reduce risk to an acceptable level. Instructors and campus network administrators must work jointly to implement controls. This paper provides an overview of controls that can protect students, instructors, and institutions of higher learning and integrates the concepts into an easy-to-use matrix to help all stakeholders ensure safety and recognize risks.

4. CONTROLS

The major types of information security controls are technical, operational, and management (Baker & Wallace, 2007). These controls can have preventive, detective, corrective, or deterrent goals. A combination of these controls is necessary for optimal risk reduction, a concept known as defense-in-depth (Butler, 2002). To help educators effectively mitigate risk, identify controls that have or could be implemented, and to recognize any risks that have been accepted, administrators and educators should evaluate the institutional controls in each cell in the matrix provided in Figure 1. Each of the types and goals of controls are discussed and more details on controls are integrated into an extended matrix in Appendix A.

	Technical	Operational	Management
Preventive	✓	✓	✓
Detective	✓	✓	✓
Corrective	✓	✓	✓
Deterrent	✓	✓	✓

Figure 1: Abridged Control Matrix

Control Types

Technical controls "focus mainly on protecting an organization's [information and communications technologies] and the information flowing across and stored in them" (Baker & Wallace, 2007, p. 37). Examples of technical controls include network and host-based firewalls, intrusion prevention systems, antivirus, and authentication. Several technical controls can be employed to protect organizations from harm. An advantage of technical controls is that they work continuously without the need for human intervention.

Operational controls are specific actions that must be carried out by personnel to proactively protect against harm or correct deficiencies (Baker & Wallace, 2007). A major difference between operational and technical controls is that operating controls are performed by people, not information systems. Vigilance is required to ensure that operational controls are being carried out properly. Examples of operational controls include awareness training, performing backups, and using secure passwords. Operational controls are typically carried out on a frequent, regular basis.

Management controls involve assessment and planning activities. Examples of management controls in a cybersecurity exercise context include performing risk assessments and vulnerability assessments. Compared to operational controls, management controls are conducted less frequently. Management controls may need to be employed whenever a major change is made to a system. Audits conducted annually help ensure that existing controls are being conducted properly.

Control Goals

Controls can have four major goals: preventive, detective, corrective, or deterrent. Preventive controls stop an event from occurring or mitigate the fallout from an event that takes place (Ko et al., 2011). Examples of preventive controls include student training, network hardening, network segmentation, and intrusion prevention systems.

Detective controls are put in place to discover when an adverse security event takes place. Detective controls are critical because "there is no absolute security that will completely prevent intrusions" (Cavusoglu, Mishra, & Raghunathan, 2004, p. 88). Examples of detective controls include intrusion detection systems, network traffic monitoring, and simply being aware of activities occurring in the classroom. Students will sometimes admit to mistakes that could have caused harm.

Corrective controls decrease the impact of an exploited vulnerability (Jones & Rastogi, 2004). It is hoped that the need to carry out corrective controls is rare, but failure to implement corrective controls can have severe consequences on the confidentiality, integrity, and availability of systems. Examples of corrective controls include backups, removing inappropriate access, and intrusion prevention systems that modify the computing environment to prevent access.

Deterrent controls attempt to prevent bad behavior "out of the perceived threat or fear of the inherent elements of sanctions" (Gopal & Sanders, 1997, p. 31). Examples of deterrent controls that can be employed in the context of cybersecurity exercises include threats of academic probation, impacts on grades, revocation of network privileges, and the possibility of legal action.

The following section describes how the risk assessment process should be driven in a cybersecurity curriculum context.

5. OPERATIONALIZATION

The burden is upon instructors to drive the risk assessment process. Instructors generally have a great deal of latitude in how they create and deliver course content to reach learning objectives. Academic freedom is one of the main drivers for choosing a career in academia (Searls, 2009). When developing exercises, instructors should follow a process for ensuring that the curriculum maximizes student success while minimizing institutional risk.

First, program curriculum and learning objectives should be defined. Syllabi should be evaluated to find activities that contain risk. For each of those activities a risk assessment should be conducted.

In conducting risk assessments, several university roles may need to be included. The instructor will be required in all cases. Programs may have dedicated lab administrators who should participate in the risk assessment when the activities impact the lab environment. Department heads and/or deans should also be included at some level, though different departments will function in idiosyncratic ways. University network administrators should be included where appropriate. Where greater risks exist, instructors should work with higher level administration. Risk management departments would need to be consulted for only the most serious risks. In some cases, even the president should be aware of risks and be asked to provide support for conducting certain exercises.

The instructor will perform the majority of the risk assessments. In many cases, instructors can implement controls themselves without needing to include others in the risk assessment process. As the risk increases or instructors are unable to implement controls, additional parties must be brought into the risk assessment process. Figure 2 shows a risk pyramid. The size of the pyramid level roughly indicates the time and effort needed to assess cyber security assessment risk.

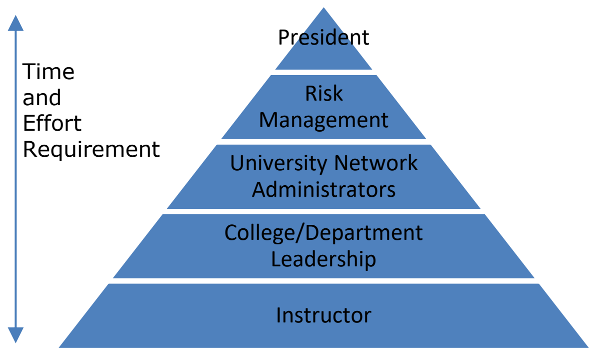

Figure 2: Risk Pyramid

To help instructors and organizations, Appendix 1 provides a sample matrix with controls that may need to be put in place depending on risk assessment outcomes. The matrix should be used as a starting point for identifying controls—not a checklist. Depending on curriculum, risks identified, risk appetite, and other factors, additional controls may be needed and some of the controls may not be necessary. However, every cell of the table should be considered carefully – from technical-preventive to management-deterrent.

An example of this process in action may be illustrative. The first time we planned to teach a course that included penetration testing concepts, we thought it would be a great experience for students to evaluate the security of local organizations. We built a relationship with one local non-profit organization and gained their buy-in. Our students and the non-profit organization were excited about the prospect of working together. As educators, we knew the students would gain valuable experience. Before beginning the engagement with the client, we drafted a legal contract with the assistance of a faculty member lawyer. The director of the non-profit was willing to sign the contract. The instructors were not legally able to sign the contract because they were not authorized agents of the university. The contract would have to be signed by a university vice president after review by the risk management department. After reviewing the scope of services proposed to conduct the security audit, the university risk management department decided the risks were too high to proceed. Despite eager students, organizations, and instructors, the project could not go forward because of risk management concerns.

The preceding story should not be considered a failure. In fact, we feel that the risk assessment process worked as intended. Risk management administrators were aware of risks and potential harm that could befall the university if something went awry. Instead of performing hands-on security evaluations of the non-profit organization, managers of the organization were invited to speak to students about the challenges they face. Virtual environments were constructed within the university to mimic the organization's infrastructure insofar as possible. Students were able to gain some of the experience they needed and learn about real-world challenges while reducing risks to the university.

6. CONCLUSIONS

No single technology or practice is the solution to the challenges of developing effective cybersecurity exercises, but whatever the exercise, care must be taken to protect the organizations from risks inherent to cybersecurity exercises. This paper provides high-level, practical guidance for organizations creating cybersecurity programs, courses, and exercises.

Some risk of harming confidentiality, integrity, or availability of systems will exist irrespective of the exercise platform. Students can make mistakes. Students can also be malicious and intentionally cause harm to systems. Risk assessments should be performed for each exercise and consider both intentional and unintentional harm that could occur. Technical, operational, and management controls with the goal of preventing, detecting, correcting, or deterring harm need to be established based on the risks identified.

The world is in dire need of qualified cybersecurity professionals. Educational institutions are working quickly to create curriculum to prepare students for challenging and exciting careers in cybersecurity. Following the guidance in this article, instruction designers can create safe cybersecurity exercises to give students the skills they need to succeed.

7. ACKNOWLEDGEMENTS

The authors would like to acknowledge the large community of software developers, educators,

and researchers working together to promote cybersecurity education.

8. REFERENCES

Baker, W., & Wallace, L. (2007). Is Information Security Under Control?: Investigating Quality in Information Security Management. *IEEE Security and Privacy Magazine*, *5*(1), 36–44. https://doi.org/10.1109/MSP.2007.11

Bernstein, C. (2013). IT Skills Shortage: The Other Critical Cliff Facing Enterprises. *EWeek*. Retrieved from http://www.eweek.com/it-management/it-skills-shortage-the-other-critical-cliff-facing-enterprises

Butler, S. A. (2002). Security Attribute Evaluation Method: A Cost-benefit Approach (p. 232). ACM Press. https://doi.org/10.1145/581339.581370

Cappelli, D., Moore, A., & Trzeciak, R. (2012). *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)* (1st edition). Upper Saddle River, NJ: Addison-Wesley.

Cavusoglu, H., Mishra, B., & Raghunathan, S. (2004). A Model for Evaluating IT Security Investments. *Communications of the ACM*, *47*(7), 87–92. https://doi.org/10.1145/1005817.1005828

Dhillon, G., & Torkzadeh, G. (2006). Value-focused assessment of information system security in organizations. *Information Systems Journal*, *16*(3), 293–314. https://doi.org/10.1111/j.1365-2575.2006.00219.x

Fidler, D. P. (2016). The U.S. Election Hacks, Cybersecurity, and International Law. *AJIL Unbound*, *110*, 337–342. https://doi.org/10.1017/aju.2017.5

Gopal, R. D., & Sanders, G. L. (1997). Preventive and Deterrent Controls for Software Piracy. *Journal of Management Information Systems*, *13*(4), 29–47.

Harrison, J. V. (2005). Enhancing Network Security By Preventing User-initiated Malware Execution (Vol. 2, pp. 597–602). Presented at the 2005 International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA: IEEE. https://doi.org/10.1109/ITCC.2005.146

Jewer, J., & Evermann, J. (2015). Enhancing Learning Outcomes through Experiential Learning: Using Open-Source Systems to Teach Enterprise Systems and Business Process Management. *Journal of Information Systems Education*, *26*(3), 187–201.

Jones, R. L., & Rastogi, A. (2004). Secure Coding: Building Security into the Software Development Life Cycle. *Information Systems Security*, *13*(5), 29–39. https://doi.org/10.1201/1086/44797.13.5.20041101/84907.5

Ko, R. K. L., Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q., & Lee, B. S. (2011). TrustCloud: A Framework for Accountability and Trust in Cloud Computing (pp. 584–588). Presented at the 2011 IEEE World Congress on Services, Washington, DC, USA: IEEE. https://doi.org/10.1109/SERVICES.2011.91

Lyon, G. F. (2008). *Nmap Network Scanning: Official Nmap Project Guide to Network Discovery and Security Scanning* (1st ed). Sunnyvale, CA: Insecure.Com, LLC.

McGrath, J. E., Martin, J., & Kulka, R. A. (1982). Dilemmatics: The Study of Research, Choices and Dilemmas. In *Judgment calls in research* (pp. 69–102). Sage Publications Thousand Oaks, CA, USA.

Nurse, J. R. C., Buckley, O., Legg, P. A., Goldsmith, M., Creese, S., Wright, G. R. T., & Whitty, M. (2014). Understanding Insider Threat: A Framework for Characterising Attacks (pp. 214–228). Presented at the 2005 International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA: IEEE. https://doi.org/10.1109/SPW.2014.38

Osborne, C. (2012, February 8). Keylogging Student Caught Hacking College Grades. Retrieved June 13, 2017, from http://www.zdnet.com/article/keylogging-student-caught-hacking-college-grades/

Sauls, J., & Gudigantala, N. (2013). Preparing Information Systems (IS) Graduates to Meet the Challenges of Global IT Security: Some Suggestions. *Journal of Information Systems Education*, *24*(1), 71–73.

Searls, D. B. (2009). Ten Simple Rules for Choosing between Industry and Academia. *PLoS Computational Biology*, *5*(6), e1000388. https://doi.org/10.1371/journal.pcbi.1000388

Verizon. (2016). *2016 Data Breach Investigations Report*. Retrieved from http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/

Wagner, D. (2016). Infrastructure Under Attack. *Risk Management*, *63*(8), 28–30.

Appendices and Annexures

Appendix A – Additional Tables and Figures

	Technical	Operational	Management
Preventive	Network Segmentation Firewalls, Network and Host-based Antivirus Authentication / Authorization Least Privilege	Physical Access Controls Training Access Review Asset Management	Risk Assessment Clean Desk Policy
Detective	Intrusion Prevention Systems Network Monitoring	Log Auditing Intrusion Detection Systems	Vulnerability Assessment
Corrective	Authorization	Backup and Restore	Business Continuity Planning Disaster Recovery Planning Incident Response Policy
Deterrent	Warning Messages	Course Policies Acceptable Use Policies (Campus and Course) Training	Disciplinary Process

Table 1: Control Matrix

Glossary of Controls

Acceptable Use Policy. All campuses should have policies that dictate appropriate use of the network. Ambiguities in the acceptable use policy do a disservice to students, instructors, and network administrators. Course policies should make clear expectations for students as to the appropriate use of security tools and techniques. Even though students are bound by public laws and campus network acceptable use policies, it is important to reinforce acceptable behavior in cybersecurity courses.

Access Review. Groups and individuals are granted access to computing resources through access control lists (ACLs). Access to resources should be periodically audited to ensure that no unnecessary access has been granted. Examples of resources that should be audited are servers, files shares, and access to administrative accounts.

Antivirus. All students on campus should run antivirus. Some cybersecurity exercises involve the analysis or creation of malware. Antivirus minimizes the risk of malware spreading across campus. Many universities choose to license antivirus so that students and faculty can install the software on their personally owned machines.

Asset Management. Computing equipment such as servers, laptops, routers, and switches should be tracked. Responsibility for maintaining that hardware must be clear.

Authentication. Campus servers, networking equipment, and other devices should be secured with strong passwords. Where appropriate, multifactor authentication should be used.

Authorization. Authorization refers to a legitimate access to a resource. When abuse occurs, authorization must be revoked. Authorization should follow the principle of least privilege—that users and services should be granted the minimal level of access required.

Backup and Restore. Data, network configurations, server configurations, and other critical data should be backed up. The backups should periodically be tested to ensure that a restore is possible.

Business Continuity Planning. Plans should be in place to ensure that the campus can still function appropriately in the case of a major outage. The business continuity plan details how work will be carried out without the use of affected computer systems.

Clean Desk Policy. People with access to sensitive information such as personal information or administrative credentials must ensure that the information is properly protected. Sensitive information should be safeguarded using locked doors and cabinets where appropriate. Multiple control points may be needed depending on the sensitivity of the information. Passwords should not be written down and kept in unguarded locations.

Disaster Recovery Planning. In case of major system outages, a disaster recovery plan details how functionality will be restored. Included in this plan is the recovery point objective and recovery time objective. The recovery point objective defines the acceptable amount of data that will be lost when restoring a system. For example, a system may only be backed up at night, so any work done after the last full backup may be lost. The recovery time objective defines the acceptable duration of restoration activities. Complex systems could take days or weeks to rebuild.

Disciplinary Process. Processes for enforcing written policies must be in place. Instructors often only have authority to enforce discipline in their classes. Discipline that exceeds classroom authority may need to be enforced by the Dean of Students.

Firewalls, Network. Network firewalls use rules to determine if network traffic can enter or leave a network segment. Network firewalls are often installed on the perimeter of a computer network. Network firewalls may also be placed between critical network segments for compliance reasons, such as protecting the cardholder data environment (CDE) for Payment Card Industry (PCI) compliance.

Firewalls, Host-based. Host-based firewalls can be enabled on individual computing devices for protection against malicious traffic. Most modern operating systems come with host-based firewalls installed and enabled by default.

Intrusion Detection System. Network administrators typically employ intrusion detection systems at the network perimeter to detect attack threats from external parties. Network administrators should consider placing intrusion detection systems where they can detect internal network traffic to identify misuse (unintentional or otherwise).

Incident Response Policy. An incident response policy defines the procedures to be carried out when a security incident takes places. The policy should include the individuals who are notified, responsibilities for communicating information about the event, and the procedures system administrators should conduct after an incident.

Intrusion Prevention System. Like intrusion detection systems, intrusion prevention systems identify malicious network traffic. Intrusion prevention systems go one step further and make changes to network configurations in an attempt to stop malicious traffic. For example, an intrusion prevention system might automatically block an IP address sending malicious traffic.

Log Auditing. Signs of hacking attempts can often be seen in computer logs. Log files should be audited to find evidence of hacking attempts, successful or unsuccessful. Log files are too large and complex to be analyzed manually. The amount of Security information and event management (SIEM) tools aid in processing logs.

Network Monitoring. Network administrators should monitor and log network traffic. Unusual system usage, such as extremely high bandwidth usage, should be questioned.

Network Segmentation. Cybersecurity exercises with the potential of causing harm may be conducted in an isolated network environment. Isolating the network prevents malicious traffic from reaching an unintended target. This can be accomplished physically (by using a separate network switch and cables), through configuration (by setting ports on a switch to an isolated VLAN), or virtually (by using virtual machines connected to an unrouted virtual network).

Physical Access Controls. Following industry best practices, telecom closets and data centers should be locked. Access to infrastructure should only be granted to authorized administrators.

Training. Network administrators should receive ongoing training that include content being taught in cybersecurity classes. Relevant topics include ethical hacking, penetration testing, and risk assessments. Because many campuses employ students, it is critical that student administrators are trained to deal with security incidents. At the beginning of teach course, students should be asked to provide assurance that they will obey all laws and abide by a standard code of ethics. This assurance can be recorded in a learning management system. Requiring students to give explicit assurance will encourage safe practices by the students and will provide the instructor some defense against organizational fallout if a student chooses to disregard course policies.

Risk Assessment. While instructors should assess the risk of individual cyber security exercises, network administrators should assess the risk to systems overall.

Vulnerability Assessments. Network administrators should periodically assess the network for vulnerabilities. Going further, penetration tests should be performed on a limited basis to ensure critical infrastructure is protected.

Warning Messages. Banner messages at login or other appropriate times can be configured to remind users about acceptable use policies and repercussions for violations.

Sprint, then Fly: Teaching Agile Methodologies with Paper Airplanes

Mark Frydenberg,
mfrydenberg@bentley.edu

David J. Yates
dyates@bentley.edu

Computer Information Systems Department
Bentley University
Waltham, Massachusetts

Julie S. Kukesh
julie.kukesh@mendix.com
Mendix Corporation
Boston, Massachusetts

Abstract

As industry embraces the agile methodology for application development, universities are shifting their curricula to teach agile principles along with traditional waterfall concepts. This paper describes a simulation game offered to students in a first-year computing concepts course to introduce both models of application development. Students work in development teams to design, build, and test paper airplanes following both waterfall and agile principles to experience the roles, processes, and challenges of each. Participants track their team's progress throughout the activity, so they can draw conclusions about the benefits and challenges of each approach. Survey results indicate that students learned the various roles and approaches of both methods through this experience.

Keywords: agile, waterfall, simulation games, app development

1. INTRODUCTION

The widespread implementation and acceptance of agile methodologies in industry during recent years has caused universities to reexamine approaches for teaching software development concepts across the curriculum. The agile methodology (Cunningham, Principles behind the Agile Manifesto, 2001) is a set of principles that define the people, process and work-output for the development and delivery of software and applications. In contrast to a traditional waterfall development methodology, which relies on gathering user requirements, developing, testing, and deploying software sequentially, agile makes use of cross-functional teams that collaborate closely with business stakeholders, flexible and progressive requirements gathering, and early delivery of working product followed by rapid iteration of these tasks.

Both agile and waterfall methodologies try to bring the right resources to the project at each phase of development. Waterfall is organizationally agnostic but siloed with respect to how to accomplish this, while agile leverages collaborative, cross-functional teams in every phase of development. The Agile Manifesto (Cunningham, Manifesto for Agile Software

Development, 2001) and guiding principles (Cunningham, Principles behind the Agile Manifesto, 2001) shown in Appendix 1 describe the tenets of the agile development process.

This paper presents an interactive game in which students build paper airplanes to simulate waterfall and agile development processes. Students learn about waterfall and agile models for software development, team member roles, and development processes that guide each methodology. The authors facilitated this exercise with students enrolled in an introductory technology course at a business university.

Organization

The paper presents a high-level overview of waterfall and agile methodologies, and summarizes relevant literature regarding the teaching of these in a variety of higher educational contexts. A discussion of simulation games used to teach agile concepts follows. The paper then describes a simulation game using paper airplanes to introduce agile concepts, and summarizes the results of this activity. The paper concludes with reflections from students and observations from the activity's facilitators.

Research Questions

The following questions guided the development of the simulation game described in this paper, and the study of its implementation:

- After participating in the simulation, will students be able to distinguish between waterfall and agile development concepts and processes?
- After completing the simulation, will students be able to describe the different team member roles, development approaches of waterfall and agile methodologies, and their benefits and limitations?
- After reflecting on the simulation, will students be able to identify how they might apply agile concepts beyond software development, into the activities of their own daily lives?

2. WATERFALL AND AGILE OVERVIEW

Figure 1 illustrates the differing approaches of waterfall and agile methodologies. Waterfall follows a sequential process, whereas the agile development process relies on frequent meetings with all participants who set their immediate goals and identify obstacles toward making progress.

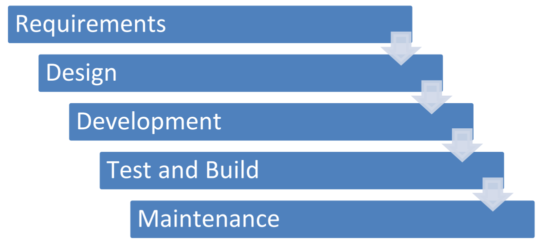

Figure 1 (a). The waterfall model is a sequential design process.

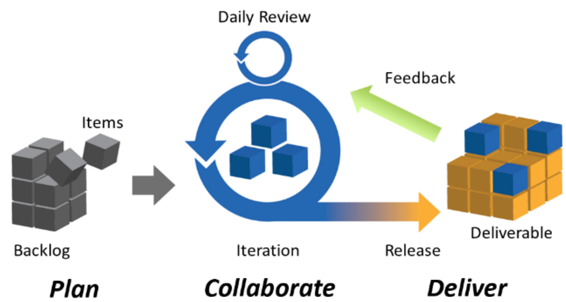

Figure 1 (b). The agile model is an iterative design process. (Fongamanda, 2012).

Both waterfall and agile methodologies place participants in roles that contribute toward the design and development of a completed product. Tables 1 and 2 present the roles implemented in many waterfall and agile development activities. The authors have simplified the names and descriptions for the sake of introductory students; agile teams in industry may use slightly different names or descriptions. In a waterfall model, participants take on the following roles, shown in Table 1.

Role	Description
Analyst	Business and systems analysts ensure that the product fulfills customer requirements and is delivered efficiently and effectively.
Developer	Responsible for design and development of product components required in the current release.
Tester	Verifies that the product is fit for customer use and validates that the product works as designed.

Table 1. Waterfall development team member roles and descriptions.

In an agile model, team members take on the following roles shown in Table 2.

Role	Description
Business Analyst	Responsible for articulating customer requirements - focusing on delivering value early and often - as well as release planning for the product.
Product Owner	Reconciles and harmonizes stakeholder requirements to help develop user stories (see Table 3) and assures the quality of the product.
Scrum Master	Owns the agile process for their team. Responsible for delivering user stories for the current iteration and coaches the team in doing so.
Developer	Merges the developer and tester roles from waterfall. Focuses on delivering components of the product needed in the current iteration.

Table 2. Agile roles and descriptions.

In addition, understanding the following vocabulary and concepts are key to participating in an agile development exercise:

Concept	Definition
User Stories	Express the "why, what and who" of desired business value for many types of product backlog items, especially features.
Product Backlog	Prioritized list of the items - features and other capabilities - needed to develop a successful product.
Standup	Short daily meeting in which team members describe what they did the day before, what they will do that day, and where they need help.
Sprint	Current iteration of product delivery, which has four activities: planning, execution, review and retrospective.

Table 3. Agile vocabulary and concepts. (Rubin, 2012).

3. AGILE ADOPTION IN THE ENTERPRISE

Implementing agile in different domains (including IT) requires a customer to whom an agile development team is going to deliver a product. Agile's methodology guides the team's activities to maximize the product's value delivered to the customer. Thus, delivering value to the customer, however defined, becomes the central mission of the team.

Agile teams are organized to provide maximum value to the business in as little time as possible. Teams rely on teamwork, focus, and collaboration. Specifically, having an agile team continuously focus on and produce value for a customer is an important measure, but not the only measure, of the team's effectiveness. It is also important to consider efficiency for a variety of reasons, not the least of which is that any team in any context has finite resources to devote to a project. In higher education, agile projects often span a semester or a quarter and deliver products usually developed by fixed-sized teams.

Companies and government agencies are adopting agile development strategies because of their focus on delivering value and their catalytic effect on innovation. According to Rigby et al. (Rigby, Sutherland, & Takeuchi, 2016, para. 7) "these days most companies operate in highly dynamic environments. They need not just new products and services but also innovation in functional processes, particularly given the rapid spread of new software tools. Companies that create an environment in which agile flourishes find that teams can churn out innovations faster in both those categories." However, these authors caution "Agile is not a panacea. It is most effective and easiest to implement under conditions commonly found in software innovation: The problem to be solved is complex; solutions are initially unknown, and product requirements will most likely change; the work can be modularized; close collaboration with end users (and rapid feedback from them) is feasible; and creative teams will typically outperform command-and-control groups." (Rigby, Sutherland, & Takeuchi, 2016, para. 12)

Despite its widespread use and professed simplicity, most organizations experience challenges when adopting agile development practices. The most recent *State of Agile* survey (VersionOne, 2017, p. 2) found that "While 94% of respondents said their organizations practiced agile, they also stated that more than half (60%) of their organizations' teams are not yet practicing agile. Similarly, although 44% of respondents stated that they were extremely knowledgeable regarding agile development practices, 80% said their organization was at or below a 'still maturing' level."

As agile adoption has increased in the enterprise (Dingsøyr, Nerur, Balijepally, & Moe, 2012), university curricula are evolving (Babb, Hoda, & Nørbjerg, 2014; Lang, 2017) to meet this market shift. In order for the next generation of technology professionals to join these organizations and contribute in a meaningful way, universities need to keep pace by teaching agile methodology to both introductory and advanced students as part of their technology curriculum.

4. TEACHING AGILE THROUGH SIMULATION GAMES

Agile methodologies have many implications in educational settings. Even though agile methodologies are most often associated with software development, many educators have taught agile concepts using scenarios or problems from a variety of domains through simulation games.

Simulation games are a common pedagogy to introduce the roles and concepts of agile methodologies in both industry and academic settings. Wangenheim, Savi, & Borgatto (2013) describe a pencil and paper game to introduce agile concepts in undergraduate courses. Players take on roles of team members, Scrum master, and product owner, as they try to create objects from paper sheets during several sprints. (Paasivaara, Heikkila, Lassenius, & Toivola, 2014) and (Krivitsky, 2017) present Scrum simulations to introduce the events and concepts of agile development by simulating sprints while building models using LEGO™ Building blocks. (Beale, 2016) and (Fernandes & Sousa, 2010) developed a board and card games to introduce scrum-based concepts by having students describe solutions to complete a task and allocate resources available to do so over several rounds. In each scenario, students need to "follow the best practices of software engineering in order to avoid any obstacle[s]" (Fernandes & Sousa, 2010, p. 53) and reach a successful outcome. The game "makes players come face-to-face in the same space to make decisions about the game, acting as an antecedent to the role of the scrum meeting in agile project management. In this way, the scrum will be easier to identify for the players/students, and the instructor can use these moments to discuss how making decisions in the game mirrors the way that scrums work organizationally." (Beale, 2016, p. 4)

Paper Airplane Simulation

This section describes the learning objectives, delivery, and results of a simulation designed to present waterfall and agile methodology and concepts to first year students enrolled in IT 101, an introductory technology and computing concepts course at a business university. The authors adapted an activity developed by ScrumInc., used as part of certified Scrum Master class training, for this classroom exercise. The goal: "Plan, build, and test as many paper airplanes as you can in 3 minutes" (Hegarty, 2013, para. 1) following the methodologies of waterfall and agile development.

Students enrolled in five different sections of IT 101 with two different instructors, participated in the simulation, which took place in an 80-minute classroom session. The same facilitators, a university alumnus and his colleague who are now working in industry at Mendix, a company whose rapid application development platform facilitates agile methodologies through the entire application lifecycle, presented the exercise to all sections on two different days, as the instructors monitored student behavior and provided encouragement.

Learning Objectives

The instructor and industry facilitators set out to create a simulation game for introductory technology students (who have little or no prior software development experience) with the following learning objectives. After completing this simulation game, students will be able to:

- Compare processes of waterfall and agile development;
- Describe tasks, concepts, and roles in traditional and agile software development processes;
- Formulate product requirements in the form of user stories that include roles, actions, outcomes, and their motivation;
- Perform roles of agile and waterfall development team members to experience benefits of each method; and
- Apply agile development principles to personal, school, and work tasks.

Presenting the Paper Airplane Activity

The facilitators began the lesson by asking students, "What would you need to do if you were going to build an app?" Students responded, "have an idea," "find someone to program it," "test it," and "publish it." Students said these tasks should happen one after the other, in this order. The facilitators pointed out the connections between the students' responses, and the waterfall methodology shown in Figure 1(a). Next, the facilitators explained to the students they were going to experience the waterfall methodology by building paper airplanes instead of developing apps.

The facilitators first briefed the students on three key components of the exercise: the objective, the roles, and the restrictions.

Objective. The objective of the waterfall model simulation is to build as many paper airplanes as possible in 12 minutes that will fly successfully over a specified distance of 15 feet when tested. The simulation evaluates a team's performance by measuring both volume (number of planes built) and quality (percentage of planes that flew the required distance). In practice, these metrics reflect the value (efficiency and quality) delivered to a customer in a development project.

Roles. Students self-organized into groups or "development teams" of three or four, and within their groups, identified the role that each student would play. One student took on the role of analyst, who would write the specifications for construction of the paper airplanes; one student took on the role of tester, who would test the paper airplanes constructed; and the remaining students acted as the development team responsible for building the paper airplanes.

This simulation exercise simplifies the various roles that are part of waterfall and agile development projects. In practice, analysts are part of a waterfall team; analysts give voice to the requirements on an agile team. A business analyst, however, generally is not a core member of an agile team.

Restrictions. The students had to use 8.5" x 11" sheets of white paper to build their airplanes. They had to test their planes on a "runway" in the center of the classroom, shown in Figure 2, or in the hallway (if the classroom layout was not conducive). The facilitators pre-measured the 15-foot distance prior to class and marked the boundaries with masking tape.

Waterfall Simulation

In the first round, which mimics a waterfall process, the facilitators gave exactly 12 minutes for students to spend 3 minutes for planning and design, 6 minutes building, and 3 minutes testing their airplanes. Students were restricted to performing work based on their assigned roles during their assigned time, and could not perform work reserved for other roles.

Each analyst wrote the requirements and design specifications for the team's airplanes during the planning and design phase and could not communicate with other team-members, nor could the analyst assist in building a prototype. The development team could begin work only during the build phase and was restricted from asking the analyst any questions or clarification on the design specifications provided to them, as well as from the results of testing any of the airplanes built. Each tester could not begin testing until the test phase, and could not make any modifications to the airplanes the development team built. The facilitators tracked time for each phase and recorded for each team the number of paper airplanes built during the build phase and the number that flew 15 feet successfully. Figure 2 shows students involved in the testing phase of the activity.

Figure 2. Students test their planes to see if they will fly a distance of 15 feet.

To end the exercise simulating a waterfall approach, the facilitators debriefed with the students about what worked and what did not work well. The debrief discussion ensured that students met the learning objectives of the traditional waterfall development approach by being able to observe and discuss the key challenges organizations experience when it comes to delivering custom applications.

Through their participation in this simulation, students recognized several application development challenges organizations may experience in this traditional waterfall approach:

- Defining requirements up front is very difficult because business requirements change;
- A lack of communication between developers and the customer may have unfavorable influence on the final product; and
- Delivering the application often takes a long time, and often the resulting product does not meet a business' expectations in the end.

Introducing Agile

Following the waterfall simulation described above, the facilitators told the students they were going to experience the agile methodology by repeating the paper airplane exercise with a different set of roles and rules that mimic the agile development process.

In the agile simulation activity, students had the same amount of time as before, 12 minutes, but split into three sprints: 1 minute for planning and design; 2 minutes to build; and 1 minute to test their airplanes. At the end of each sprint, they could go back, share their results with the team, whose members would offer the builders suggestions for improving the design prior to the next testing phase. Throughout each sprint, the facilitators recorded the number of airplanes built and the number successfully flown for each group. At the end of the three sprints, the facilitators engaged students in a similar debrief discussion where they reflected on the results of the agile version of the activity, compared and contrasted how their experience differed between the traditional waterfall methodology and agile, and compared the results of agile to those from the first activity using waterfall.

The facilitators tracked the number of planes built and flown successfully using waterfall and agile approaches. The charts in Appendix 2 Figures 1-5 and data in Appendix 2, Figure 6 summarize the results for 31 teams across five sections of IT 101. Appendix 2, Figures 1 and 2 show the number of planes built and flown successfully using waterfall and agile methodologies. The majority of teams experienced higher success rates using agile over waterfall methodologies. Appendix 2, Figure 3 shows the number of airplanes built during each of the three sprints in the agile development round of the game. For most teams, the number of planes built in successive sprints was the same or more than the number built in earlier sprints. This suggests that the collaborative process in agile had a positive impact on each team's results. Appendix 2, Figure 4 shows the number of airplanes each team built using waterfall and agile methods. For almost all of the 31 teams, the total number of airplanes built using agile methods was higher than the number built following waterfall methods.

Appendix 2, Figure 5 shows the average success rates (number of planes flown successfully divided by the number of planes built) for each team using waterfall and agile methods. Of the 31 teams, three had perfect success rates using both waterfall and agile methods; six had perfect success rates with waterfall only, and seven teams had perfect success rates using agile methods. For only five of the 31 teams did agile result in a lower success rate than waterfall. This exercise gave the students the sense that agile's iterative design process involves more people and generally produces superior results (Babb, Hoda, & Nørbjerg, 2014).

5. STUDENT REFLECTIONS

This section presents the results of a written assignment completed within three days of the simulation game, in which 76 students (42 male, 34 female) reflected on what they learned. Before the original activity, most students were not familiar with agile or waterfall methodologies as shown in Figure 3.

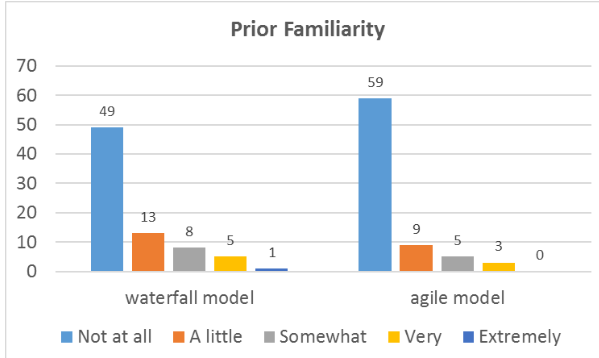

Figure 3. Prior familiarity with development methodologies.

The instructors provided three different ways for students to learn about software methodologies: reading an article introducing agile development terminology (Field, 2014) before class, attending an in-class presentation, and participating in the paper-airplane simulation activity. Figure 4 summarizes the extent to which students felt each of these contributed to their learning.

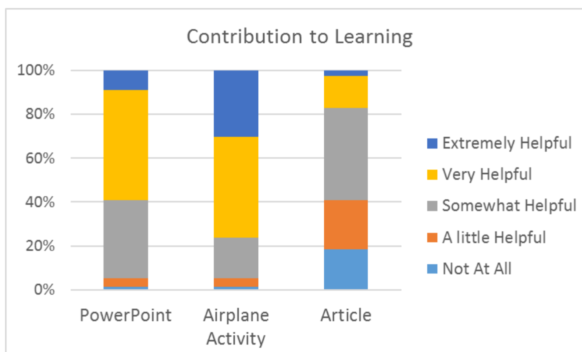

Figure 4. Contribution to learning.

Forty-five of the students thought the PowerPoint presentation was very or extremely helpful; 58 said the Airplane activity was very or extremely helpful; only 13 said the assigned article was very or extremely helpful in their understanding of agile and waterfall concepts.

When asked in a fill-in-the-blank question to identify which role, as presented in Tables 1 and 2, that they took on in each part of the simulation, most students remembered the tasks they completed, although some identified their roles with a different name, as shown in Table 4.

Role	Waterfall	Agile
Developer	24	25
Tester	20	14
Product Owner		7
Scrum Master		7
Analyst	16	6
Builder	7	5
Designer	1	1
Director		1
Don't Remember	4	5
Other	2	3
Not Present	1	1
Total	**75**	**75**

Table 4. Student roles in waterfall and agile simulations.

Developers and testers were much more likely to recall their roles correctly. Some students were confused about what to call the third role in agile: responses include builder, designer, director, product owner, Scrum master. Some responsibilities needed some clarification as well; for example, a developer develops in waterfall, but develops and tests in agile. Some students were confused that the label has a different meaning in different contexts. This may have to do with the design and presentation of the exercise.

Waterfall Reflections

During the waterfall activity, builders recalled their role as follows: "I constructed the paper airplanes as efficiently as possible;" and "I tried to build a paper airplane according to the specs that the analyst gave me." Testers said, "I had to test the product to see which ones flew 12 feet, and which ones didn't;" and "My job was to throw the airplanes at the best distance possible."

Students tried to adhere to the boundaries of their roles. Said one student in the analyst role: "I was not allowed to talk about to plan or touch the airplanes while they were being developed." A developer said, "I was not really able to change the design of the planes, or know what worked or what didn't." Another said, "It was hard to follow the directions of the analyst and make paper airplanes fast and according to the specs." "We could not communicate with the other members of our team so it was hard to build a successful airplane."

When asked about the benefits of the waterfall method, students commented, "Every member of the team had a specific focus which made the process very specialized; and "We were acting as an assembly line as the paper plane project moved through the required steps." They also recognized issues with the waterfall development method as applied to this simulation. Remarks included "It was a waste of time because everybody had to wait for the previous person to finish working;" "We were very siloed and didn't collaborate at all, which was frustrating;" and "There was attention to detail, but as the tester, I tested a finished product I had never seen before."

Agile Reflections

During the agile activity, students commented on their roles. Analysts recalled, "I had to find a model for the paper plane that was going to be built. I did these three different times after we made adjustments to the product each time to improve mistakes;" and "I was responsible for working with the other member to design a model for the airplane. Again, I drew up a more comprehensive diagram for the dev team to follow."

The activity made students aware of how the different development methods influenced their group's collaborative process and results. Comments included "The shorter times allowed us to hear from our analyst and scrum leader more so I felt like I knew what I was doing more;" "I was not able to help create the actual planes because I was the analyst and not part of the development team;" and "I did not exactly get to write up the instructions or create the airplane but was allowed to add input and collaborate with group members, so I was able to contribute much more during this process."

Regarding the benefits of the agile simulation, students commented, "We learned how to work collaboratively in a group;" "I like it better because it allowed us to see what parts we were doing wrong and enabled us to improve upon our prior attempts much easier;" "Speeding up the process let us address issues in our models quickly and efficiently;" "This was a much easier method. We were able to communicate and therefore we could tweak the plan from round to round and perfect the airplane. It helped us have a better success rate and also waste less time."

Noting difficulties of the agile process, one group stated, "Because we were more pressed for time, our quality decreased substantially. However, we tripled our production numbers;" and "I could not interfere with the analyst while she was writing the instructions even though I knew the plane she was going for was not going to work."

To answer the third research question, the facilitators asked students to consider how they might apply agile principles to other areas of their personal, school, or work lives. They commented: "When working at my internship this will come in handy and also when doing homework. Do short bursts of work and then reward yourself with a little relaxing." Another said, "I could use it to improve my studying skills. I could study for a short sprint, and test what I know. After I realize what I missed, that's what I'll focus on for the next sprint. I can continue that for as long as it takes until I know the material."

Several students noted that agile methods are a good model for improved collaboration and group work. "Increasing communication at multiple times in a products production cycle will improve the final product. This is also true in the case of a paper written by a team of five. Less iteration is necessary in a personal environment, as there is only one team member." Students reflected on the global lessons they learned from acting agile: "It is important to do your best the first time with anything, especially knowing that you always can’t go back to fix it." Another student noted, "Spending a shorter amount of time planning gives you multiple opportunities to see which models work best. I can apply this to planning and creating school projects." Another student expanded, "For life in general, it is important to break larger projects into smaller groups and discuss with people your progress on your way to your long-term goal."

Regarding the paper airplane activity, students said: "I think that the activity went very well and clearly showed the difference between the waterfall and agile method. The activity possibly could be improved by slightly increasing the number of people in each group or allowing the groups to interact more and work together more during the agile method compared to the waterfall method." Another student commented, "The activity really represented the waterfall and agile methods well. It made me understand the idea and process, [and was] good way to learn how app development companies go about creating apps and software nowadays."

6. OBSERVATIONS AND CONCLUSIONS

Students were engaged in the exercise and comprehended the basic ideas of both methodologies.

Some of the analysts used search engines during the research part to look up how to build a paper airplane in order to provide accurate instructions for their team of developers. The facilitators pointed out that using a search engine for this purpose in this context is analogous to gathering requirements accurately and documenting them prior to beginning the development process. Other analysts tried to build a prototype rather than writing the instructions and specifications.

Often, students in the analyst role conflated their roles and responsibilities with those of a developer. Many developers had a difficult time sticking to their own role and waiting their turn. Universally, the developers tried to start building before the allocated "build" time had begun and after it ended. Additionally, they were vocal with their criticism of the analyst instructions both during the activity time and during the debrief discussion.

Order of Presentation

During the first two iterations of this classroom activity, the facilitators presented the waterfall method first, followed by agile, in all sections. During the following semester, they presented waterfall first and agile second in half of the sections, and agile first/waterfall second in the other half of the sections. The goal was to see if order in which students learned about agile and waterfall methods had any impact on their results.

The facilitators found that when presenting agile first and waterfall second, the waterfall activity appeared to serve as a fourth sprint. The builders already knew how to make planes effectively based on the previous three sprints, so little room for improvement was possible. Even when the facilitators changed requirements slightly (for example, requiring the use a half-sheet of paper rather than a full sheet, or changing the flying distance), distinctions between the development phases of both models were blurred because of prior experiences.

Improvements

Based on their experience facilitating this activity, the facilitators recommend several modifications to improve it for a future offering:

- Modify the simplified agile roles as presented to reflect these roles more accurately in agile development projects in the workplace: The analyst role would collapse with the product owner role; the Scrum master would be a team lead responsible for keeping time; and the agile development team would self-organize to test and build the airplanes.

- Conduct a sprint retrospective (Rubin, 2012) after each sprint to include process, communication, and other improvements in the next iteration.
- Reorganize the activity to present, simulate, and debrief on one methodology at a time, rather than introducing both, simulating both, and then debriefing on both after the entire exercise concludes. (Facilitators implemented this change for the last two of the five sections.)
- Follow up the simulation with a simple app development activity using a tool such as Mendix, a no-code, model-driven development tool that facilitates the process of creating mobile and web apps without prior programming knowledge. Creating a working app using a development tool will allow students to apply concepts from the simulation to a real-world development project as part of an agile team without having to already know, or learn, how to program.

Conclusions

The debrief sessions at the end of each simulation reflect an agile sprint retrospective in which participants reflected on their process, communication and also the successes, and challenges, in working together. They felt very "siloed" in the waterfall simulation and much more collaborative in the agile simulation. Future iterations of this simulation might alternate introducing agile and waterfall methods first in different sections, to minimize any perceived influence that one method is superior to the other.

This lesson on introducing agile and traditional methodologies brings awareness about the process of creating apps, the roles involved in app development beyond developers, and the various tasks that each team member must complete. Through the paper airplane development simulation described here, students gained an appreciation for current software development methodologies that they otherwise might not see until later in their studies.

7. ACKNOWLEDGEMENTS

The authors thank Conner Charlebois, alumnus and Senior Solutions Consultant at Mendix Corporation, for initiating, refining, and facilitating the paper airplane simulation activity.

8. REFERENCES

Babb, J., Hoda, R., & Nørbjerg, J. (2014). Embedding Reflection and Learning into Agile Software Development. *IEEE Software, 31(4)*, 51-57.

Beale, M. (2016). Designing an Agile Game for Technical Communication Classrooms. *SIGDOC '16: Proceedings of the 34th ACM International Conference on the Design of Communication* (pp. 17:1-17:9). Silver Spring, MD: ACM.

Cunningham, W. (2001). *Manifesto for Agile Software Development*. Retrieved from Agile Manifesto: http://agilemanifesto.org/

Cunningham, W. (2001). *Principles behind the Agile Manifesto.* Retrieved from Agile Manifesto: http://agilemanifesto.org/principles.html

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A Decade of Agile Methodologies: Towards Explaining Agile Software Development. *Journal of Systems and Software, 85(6)*, 1213–1221.

Fernandes, J., & Sousa, S. M. (2010). PlayScrum - A Card Game to Learn the Agile Method. *Second International Conference on Games and Virtual Worlds for Serious Applications* (pp. 52-59). IEEE.

Field, D. (2014, November 7). *Embrace Agile Requirements Gathering And Best Practices*. Retrieved from Mendix: https://www.mendix.com/blog/an-agile-development-approach-to-requirements-gathering/

Fongamanda. (2012, May 22). *File: Agile Project Management by Planbox.png.* Retrieved from Wikimedia Commons, the free media repository: https://commons.wikimedia.org/wiki/File:Agile_Project_Management_by_Planbox.png

Frydenberg, M., & Press, L. (2009). From Computer Literacy to Web 2.0 Literacy: Teaching and Learning Information Technology Concepts Using Web 2.0 Tools. *Infomation Systems Education Journal, 8*(10), 18.

Hegarty, C. (2013, March 4). *The Serenity of Flow*. Retrieved from ScrumInc.:

https://www.scruminc.com/the-serenity-of-flow/

Krivitsky, A. (2017). *Lego4Scrum: Scrum Simulation with LEGO.* Leanpub.

Lang, G. (2017). Agile Learning: Sprinting Through the Semester. *Information Systems Educational Journal, 15(3)*, 14-21.

Paasivaara, M., Heikkila, V., Lassenius, C., & Toivola, T. (2014). Teaching students scrum using LEGO blocks. *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 382-391). Hyderabad: ACM.

Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016, May). Embracing Agile. *Harvard Business Review, 50*, 40–48.

Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process.* Boston, Ma: Addison-Wesley.

Thomas, J. D., & Blackwood, M. (2010). Computer Literacy and Non-IS majors. *Information Systems Education Journal, 8*(58), 3-12. Retrieved from http://isedj.org/8/58

VersionOne. (2017). *11th annual State of Agile Report.* Alpharetta, GA.

Wangenheim, C. G., Savi, R., & Borgatto, A. F. (2013, October). SCRUMIA - An educational game for teaching SCRUM in computing courses. *Journal of Systems and Software, 86*(10), 2675-2687.

Editor's Note:

This paper was selected for inclusion in the journal as an EDSIGCON 2017 Distinguished Paper. The acceptance rate is typically 7% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2017.

Appendix 1. Agile Manifesto and Guiding Principles

Agile Manifesto

The Manifesto for Agile Software Development (Cunningham, 2001) outlines principles for improving the process of software development:

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- *Individuals and interactions* over processes and tools
- *Working software* over comprehensive documentation
- *Customer collaboration* over contract negotiation
- *Responding to change* over following a plan

That is, while there is value in the items on the right, we value the items on the left more."

Guiding Principles

The guiding principles behind the Agile Manifesto (Cunningham, Principles behind the Agile Manifesto, 2001) are:

We follow these principles:

Our highest priority is to satisfy the customer
through early and continuous delivery
of valuable software.

Welcome changing requirements,
even late in development.
Agile processes harness change
for the customer's competitive advantage.

Deliver working software frequently,
from a couple of weeks to a couple of months,
with a preference to the shorter timescale.

Business people and developers
must work together daily throughout the project.

Build projects around motivated individuals.
Give them
the environment and support they need,
and trust them to get the job done.

The most efficient and effective method
of conveying information
to and within a development
team is face-to-face conversation.

Working software
is the primary measure of progress.

Agile processes
promote sustainable development.
The sponsors, developers, and users
should be able
to maintain a constant pace indefinitely.

Continuous attention
to technical excellence
and good design enhances agility.

Simplicity--
the art of maximizing the amount
of work not done--
is essential.

The best architectures,
requirements, and designs
emerge from self-organizing teams.

At regular intervals, the team reflects
on how to become more effective,
then tunes and adjusts its behavior accordingly.

Appendix 2. Paper Airplane Activity Results

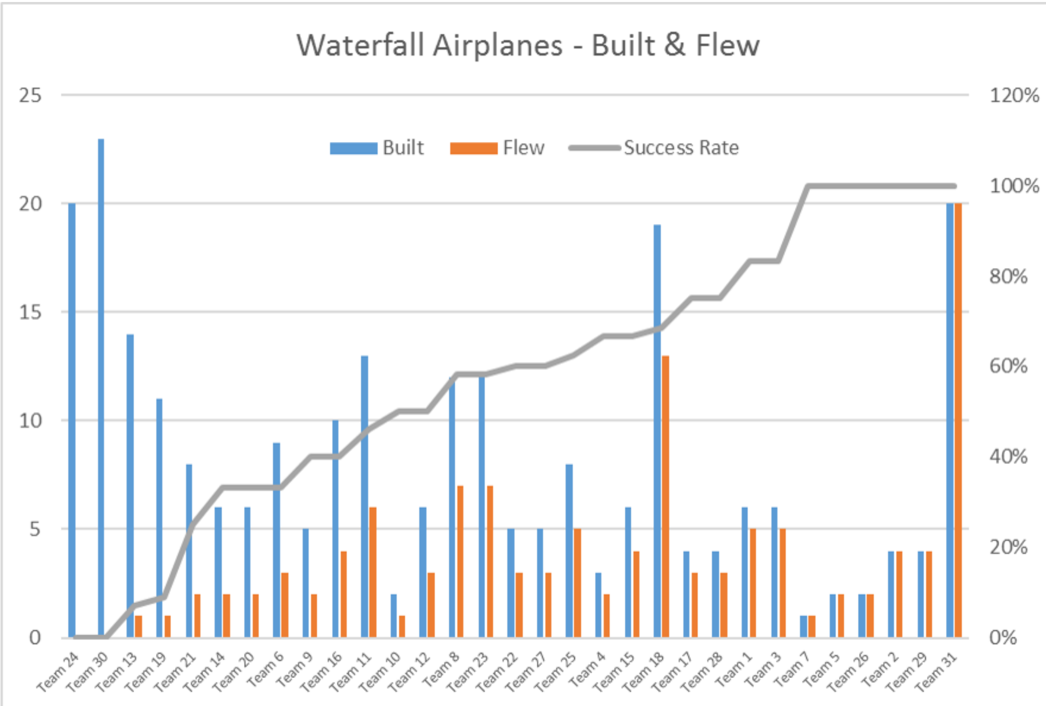

Figure 1. Airplanes built and that flew – Waterfall

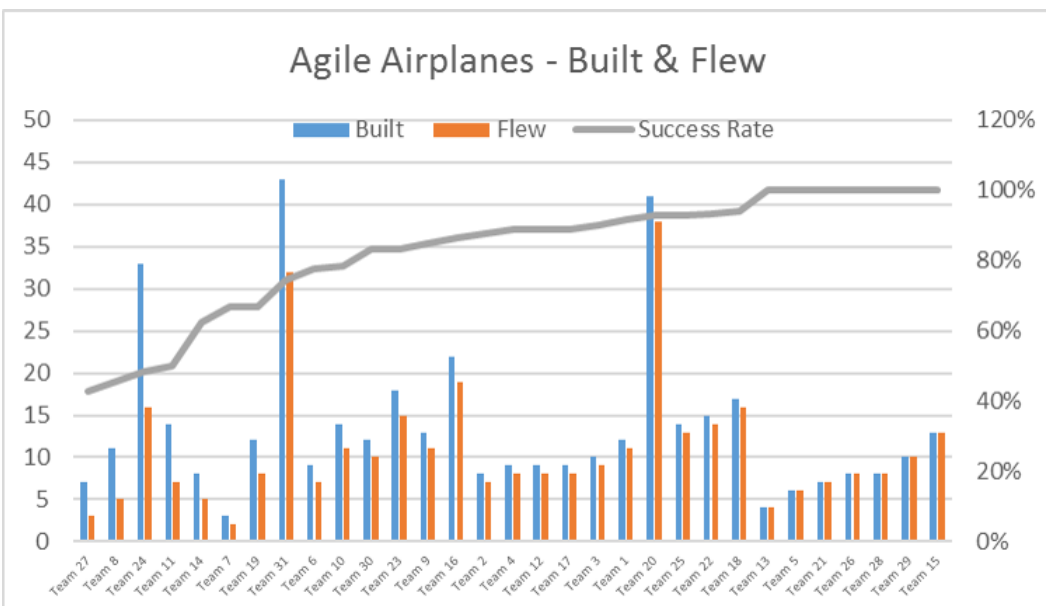

Figure 2. Airplanes built and that flew - Agile

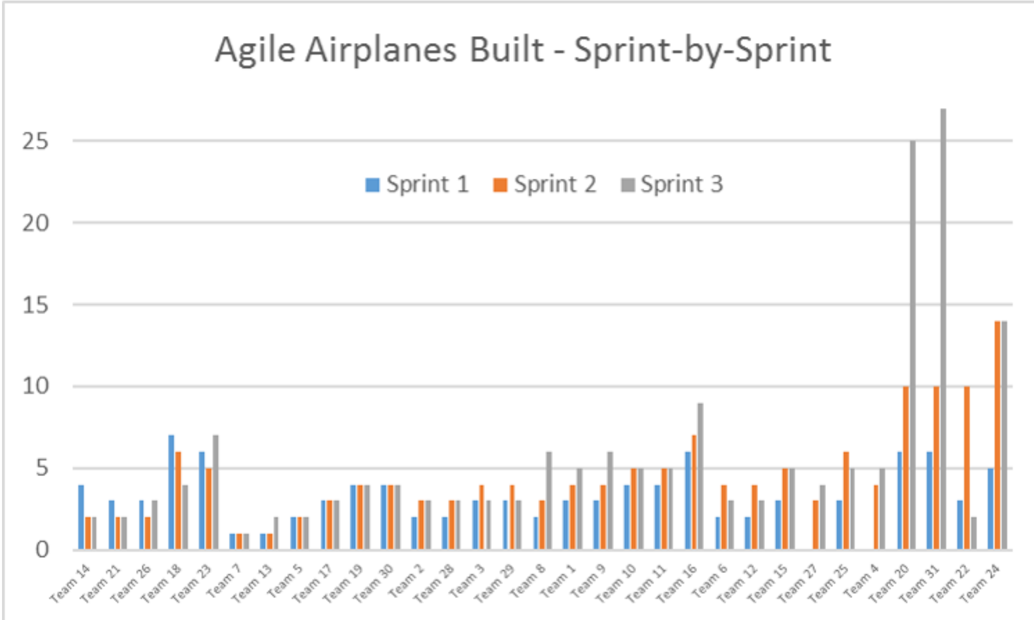

Figure 3. Airplanes built during each sprint

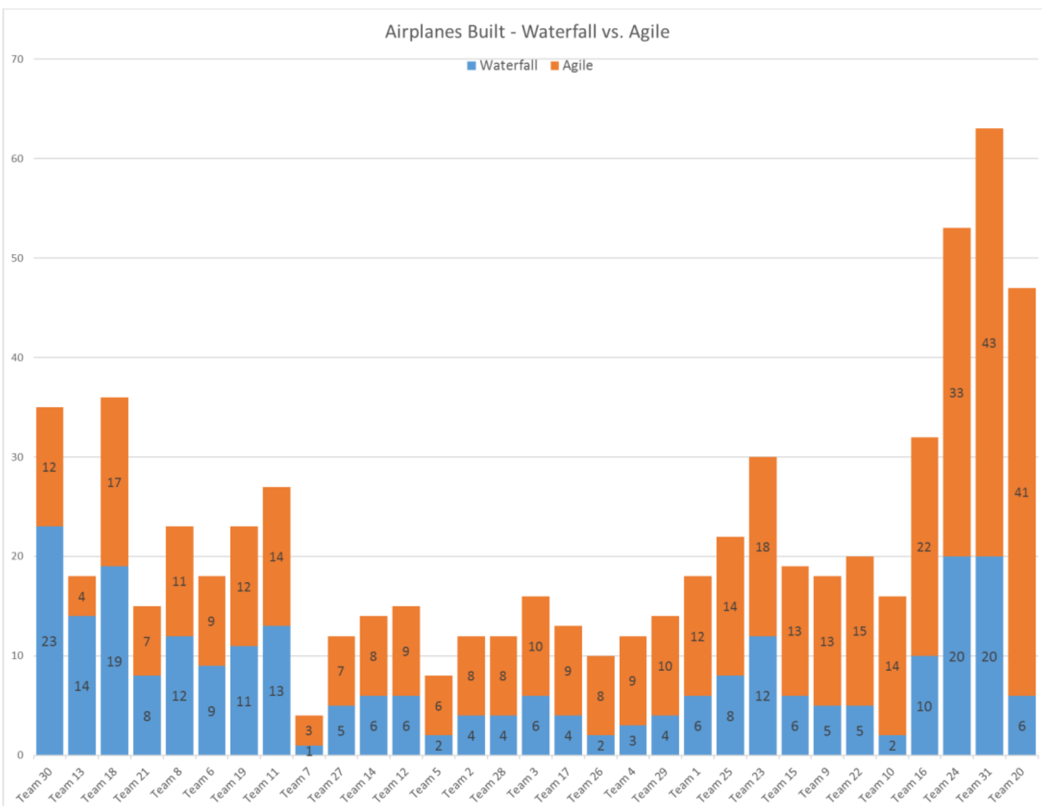

Figure 4. Number of airplanes built by each team using waterfall and agile methods

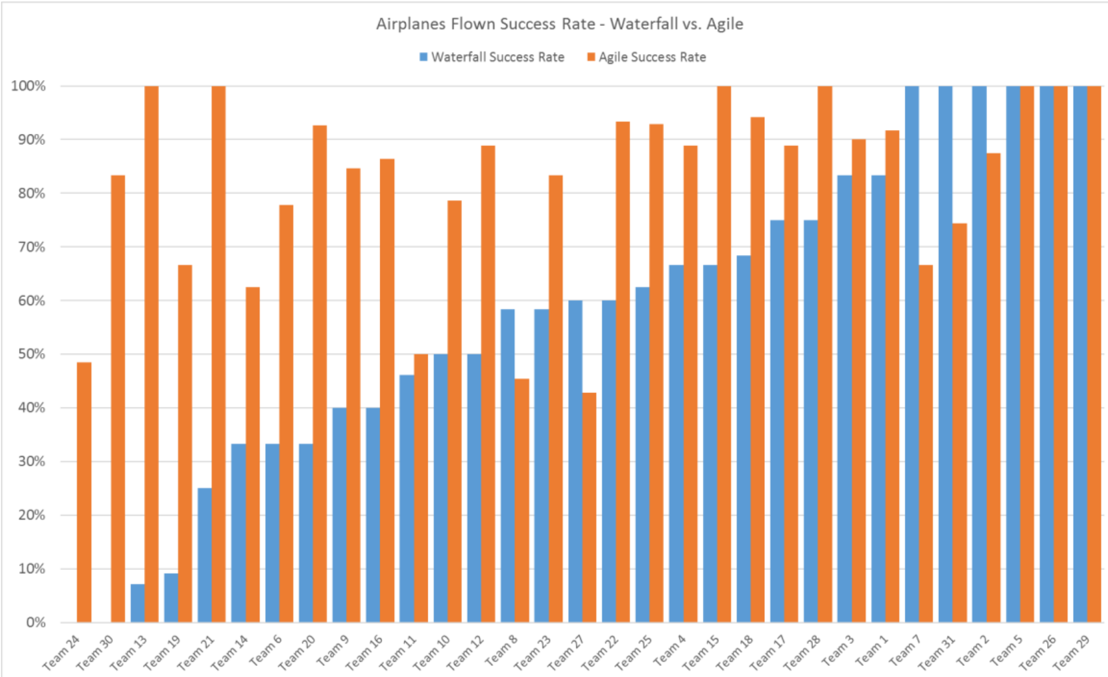

Figure 5. Increased average success rate

		31 Teams in 5 Class Sections																							
		AGILE BY SPRINT												AGILE ALL SPRINTS			WATERFALL vs. AGILE						% improvement Agile over Waterfall		
		Built				Flew				Success Rate (%)				TOTAL			Built		Flew		Success Rate				
		S1	S2	S3		S1	S2	S3		S1	S2	S3		Built	Flew	Success Rate	Trad'l	Agile	Trad'l	Agile	Trad'l.	Agile	Built	Flew	Success Rate
TEAM 1	Class A	3	4	5	**12**	3	4	4	**11**	100%	100%	80%	**92%**	**12**	**11**	**92%**	**6**	12	**5**	11	83%	92%	200%	220%	8%
TEAM 2	Class A	2	3	3	**8**	1	3	3	**7**	50%	100%	100%	**88%**	**8**	**7**	**88%**	**4**	8	**4**	7	100%	88%	200%	175%	-13%
TEAM 3	Class A	3	4	3	**10**	3	3	3	**9**	100%	75%	100%	**90%**	**10**	**9**	**90%**	**6**	10	**5**	9	83%	90%	167%	180%	7%
TEAM 4	Class A	0	4	5	**9**	0	3	5	**8**	Error	75%	100%	**89%**	**9**	**8**	**89%**	**3**	9	**2**	8	67%	89%	300%	400%	22%
TEAM 5	Class A	2	2	2	**6**	2	2	2	**6**	100%	100%	100%	**100%**	**6**	**6**	**100%**	**2**	6	**2**	6	100%	100%	300%	300%	0%
TEAM 6	Class A	2	4	3	**9**	2	3	2	**7**	100%	75%	67%	**78%**	**9**	**7**	**78%**	**9**	9	**3**	7	33%	78%	100%	233%	44%
TEAM 1	Class B	1	1	1	**3**	0	1	1	**2**	0%	100%	100%	**67%**	**3**	**2**	**67%**	**1**	3	**1**	2	100%	67%	300%	200%	-33%
TEAM 2	Class B	2	3	6	**11**	1	2	2	**5**	50%	67%	33%	**45%**	**11**	**5**	**45%**	**12**	11	**7**	5	58%	45%	92%	71%	-13%
TEAM 3	Class B	3	4	6	**13**	1	4	6	**11**	33%	100%	100%	**85%**	**13**	**11**	**85%**	**5**	13	**2**	11	40%	85%	260%	550%	45%
TEAM 4	Class B	4	5	5	**14**	2	4	5	**11**	50%	80%	100%	**79%**	**14**	**11**	**79%**	**2**	14	**1**	11	50%	79%	700%	1100%	29%
TEAM 5	Class B	4	5	5	**14**	1	3	3	**7**	25%	60%	60%	**50%**	**14**	**7**	**50%**	**13**	14	**6**	7	46%	50%	108%	117%	4%
TEAM 1	Class C	2	4	3	**9**	2	3	3	**8**	100%	75%	100%	**89%**	**9**	**8**	**89%**	**6**	9	**3**	8	50%	89%	150%	267%	39%
TEAM 2	Class C	1	1	2	**4**	1	1	2	**4**	100%	100%	100%	**100%**	**4**	**4**	**100%**	**14**	4	**1**	4	7%	100%	29%	400%	93%
TEAM 3	Class C	4	2	2	**8**	1	2	2	**5**	25%	100%	100%	**63%**	**8**	**5**	**63%**	**6**	8	**2**	5	33%	63%	133%	250%	29%
TEAM 4	Class C	3	5	5	**13**	3	5	5	**13**	100%	100%	100%	**100%**	**13**	**13**	**100%**	**6**	13	**4**	13	67%	100%	217%	325%	33%
TEAM 5	Class C	6	7	9	**22**	5	6	8	**19**	83%	86%	89%	**86%**	**22**	**19**	**86%**	**10**	22	**4**	19	40%	86%	220%	475%	46%
TEAM 6	Class C	3	3	3	**9**	2	3	3	**8**	67%	100%	100%	**89%**	**9**	**8**	**89%**	**4**	9	**3**	8	75%	89%	225%	267%	14%
TEAM 1	Class D	7	6	4	**17**	7	6	3	**16**	100%	100%	75%	**94%**	**17**	**16**	**94%**	**19**	17	**13**	16	68%	94%	89%	123%	26%
TEAM 2	Class D	4	4	4	**12**	1	3	4	**8**	25%	75%	100%	**67%**	**12**	**8**	**67%**	**11**	12	**1**	8	9%	67%	109%	800%	58%
TEAM 3	Class D	6	10	25	**41**	3	10	25	**38**	50%	100%	100%	**93%**	**41**	**38**	**93%**	**6**	41	**2**	38	33%	93%	683%	1900%	59%
TEAM 4	Class D	3	2	2	**7**	3	2	2	**7**	100%	100%	100%	**100%**	**7**	**7**	**100%**	**8**	7	**2**	7	25%	100%	88%	350%	75%
TEAM 5	Class D	3	10	2	**15**	2	10	2	**14**	67%	100%	100%	**93%**	**15**	**14**	**93%**	**5**	15	**3**	14	60%	93%	300%	467%	33%
TEAM 6	Class D	6	5	7	**18**	4	4	7	**15**	67%	80%	100%	**83%**	**18**	**15**	**83%**	**12**	18	**7**	15	58%	83%	150%	214%	25%
TEAM 7	Class D	5	14	14	**33**	2	14		**16**	40%	100%	0%	**48%**	**33**	**16**	**48%**	**20**	33	**0**	16	0%	48%	165%	Error	48%
TEAM 1	Class E	3	6	5	**14**	2	6	5	**13**	67%	100%	100%	**93%**	**14**	**13**	**93%**	**8**	14	**5**	13	63%	93%	175%	260%	30%
TEAM 2	Class E	3	2	3	**8**	3	2	3	**8**	100%	100%	100%	**100%**	**8**	**8**	**100%**	**2**	8	**2**	8	100%	100%	400%	400%	0%
TEAM 3	Class E	0	3	4	**7**	0	2	1	**3**	Error	67%	25%	**43%**	**7**	**3**	**43%**	**5**	7	**3**	3	60%	43%	140%	100%	-17%
TEAM 4	Class E	2	3	3	**8**	2	3	3	**8**	100%	100%	100%	**100%**	**8**	**8**	**100%**	**4**	8	**3**	8	75%	100%	200%	267%	25%
TEAM 5	Class E	3	4	3	**10**	3	4	3	**10**	100%	100%	100%	**100%**	**10**	**10**	**100%**	**4**	10	**4**	10	100%	100%	250%	250%	0%
TEAM 6	Class E	4	4	4	**12**	2	4	4	**10**	50%	100%	100%	**83%**	**12**	**10**	**83%**	**23**	12	**0**	10	0%	83%	52%	Error	83%
TEAM 7	Class E	6	10	27	**43**	2	3	27	**32**	33%	30%	100%	**74%**	**43**	**32**	**74%**	**20**	43	**20**	32	100%	74%	215%	160%	-26%
	ALL	100	144	175	**419**	66	125	102	**339**	**66%**	**87%**	**58%**	**81%**	**419**	**339**	**81%**	**256**	**419**	**120**	**339**	**47%**	**81%**	**164%**	**283%**	**34%**

Figure 6. Summary of Paper Airplane Exercise results across all sections

Data Analytics Workshop Series for Non-Computing Major First-Generation-College-Bound Students

Sam Chung
SamChung@siu.edu
School of Information Systems and Applied Technologies
Southern Illinois University
Carbondale, IL 62901, USA

Abstract

The purpose of this research is to propose how we can encourage non-computing major first-generation-college-bound students to be actively involved in learning data analytics. Non-computing major students have limited opportunities to take a data analytics related course. The computing major programs have the resource limit for offering none-major electives. The first-generation-college-bound students need more mentoring for their future directions. For this purpose, we challenge the following three goals: 1) What practices can we use with underpinning scientific evidences? 2) How can we enhance engagement of students with very limited interventions from an instructor? And, 3) how can we motivate the participation of the non-computing major students? To accomplish these goals, we start with developing a series of summer workshops that is Evidence-Based Practices (EBP)-guided, student-driven, and applied. Based upon the EBPs, we develop a series of student-driven summer workshops, not a regular elective course, with an emphasis on application of data analytics to the main areas in which the students are interested. We conclude that the EBP first helped us to develop these workshop series for applied data analytics with underpinning scientific evidences. Second, these workshops using active learning methods allowed the students to have their strong engagement with very limited interventions from an instructor. Third, these workshops motivated the participation of the non-computing major students by influencing the students to seek how data analytics can be applied to their domain. Last, the outcome of their team project allowed them to experience undergraduate research.

Keywords: Data Analytics, Non-Computing Major, First-Generation-College-Bound, Evidence-Based Practice, Active Learning Methods, Experiential Learning.

1. INTRODUCTION

In the era of Big Data, Data Science becomes an emerging subject across disciplines. Because of the multidisciplinary and interdisciplinary subject, it is important for students to learn concepts of data analytics, earn skills of data analysis, and then apply the skills to their real word problems regardless of their major (Wymbs, 2016; Wright, 2016).

Although students in a computing major such as Computer Science, Information Technology, and Information Systems can take a data analytics course as a regular class, most of non-computing major students have limited opportunities to take a data analytics related course. Since many computing programs have to teach pre-existing courses, it is not easy for them to open a data analytics course for non-major students. In addition, underrepresented students such as first-generation-college-bound students or socioeconomically disadvantaged students have limited resources of understanding their demands for their advanced study of professional careers in emerging areas (National Academic of Sciences (NAS), 2011; The Executive Office of the President, 2014).

To solve the constraints of non-computing major students, the resource limit of the computing major programs, the lack of mentoring of the first-generation-college-bound students, we propose an approach that focuses on first-generation-college-bound student, not instructor but student-driven and non-computing majors. However, we encounter three challenges: 1) What scientific evidence-based practices can we use? 2) How can we enhance engagement of students with very limited interventions from an instructor? And, 3) how can we motivate the participation of the non-computing major students?

For those three challenges, we start with identifying best practices that have sound scientific evidences to make a course very friendly for the students who are non-computing major and first-generation-college-bound. Based upon the identified Evidence-Based Practices (EBP), we first choose the workshop series format for a small group during summer (Sackett, Rosenberg, Gray, Haynes, & Richardson, 1996). Second, to encourage the students drive the workshop series, we infuse active learning methods and a classroom assessment technique into each workshop module. Third, to be attractive to non-computing majors, we allow students to apply data analytics to their real problems and present their peer-reviewed research outcomes at a local conference. Fourth, we assess the workshop series in terms of formative and summative evaluation. Then, we discuss what we learned through the series of summer workshops. Last, we conclude that we could encourage non-computing major first-generation-college-bound students to be actively involved in learning data analytics.

2. BACKGROUND

Evidence-Based Practice (EBP)
To select a correct format for non-computing major first-generation-college-bound students, we adopt Evidence-Based Practice (EBP), which came from Medicine. According to Dr. David L. Sackett, who is known as one of the fathers of Evidence-Based Medicine (EBM), EBM means clinical practices should be conducted based upon the best available external scientific clinical evidences (Sackett, et al., 1996; Straus, Glasziou, Richardson, Haynes, & Sackett, 2011). Software Engineering research community adopted EBM and proposed Evidence-Based Software Engineering (EBSE) (Kitchenham, Dyba, & Jorgensen, 2004; Dyba, Kitchenham, & Jorgensen, 2005).

Active Learning Methods – FC, JiTT, & PI
An active learning method allows each student to be more directly involved in his or her leaning process. Each student is a learner as connection-maker, content producer, and sharer. We use three active learning methods: Flipped Classroom (FC), Just-in-Time Teaching (JiTT), and Peer Instruction (PI).

FC is an active learning method that reverses typical lecture in classroom to homework activities in classroom (Bergmann & Sams, 2012; Rutherfoord & Rutherfoord, 2013). Students are required to study lecture materials at home by watching short videos and solving short quizzes before the class session while they are devoted to discussions, exercises, or projects in class. JiTT is an active learning method that allows an instructor to adjust lecture materials within a short time before a lecture begins (Novak, Patterson, Gavrin, & Christian, 1999; Gurka, 2012; Martinez, 2012). PI is an active learning method that encourages student to help other students by discussing an answer of a given concept test (Mazur, 1997; Mazur and Watkins 2010; Simon and Cutts, 2012). The main point of a concept test is to confirm whether the students in class truly understood one important concept that the student should take away from the class.

Classroom Assessment Techniques (CAT's)
CAT's are strategies that allow instructors to conduct formative assessment in order to assess how well students are learning key concepts during class time (Angelo and Cross, 1993). Among CAT's, we use the muddiest point that allows students to have their own chance to briefly describe what part of the lesson or the assignment in class was most confusing to them.

3. RELATED WORK

Data Science becomes an emerging subject across disciplines for both undergraduate and graduate degree programs. Data Analytics has been one of important Body of Knowledge (BoK) of Data Science. A Data Science degree program was proposed and implemented for an interdisciplinary undergraduate degree (Anderson, Bowring, McCauley, Pothering, & Starr, 2014). An interdisciplinary data analytics track and its minor for undergraduate students were proposed (Wymbs, 2016). A Master's of Science degree in Data Analytics was also proposed for business major students (Jafar, Babb, & Abdullat, 2016). A data analytic centric MS Degree was proposed for Information Sciences and Technologies students (Kang, Holden, & Yu, 2014; Kang, Holden, & Yu, 2015). An online graduate program in Information Security and Analytics was proposed (Kumar, 2014).

However, most efforts above focused on a degree program with regular courses for either undergraduate or graduate students. The degree program considered multidisciplinary and interdisciplinary subjects. Some degree programs emphasized applied data analytics with advanced skills (Kang, Holden, & Yu, 2014; Kang, Holden, & Yu, 2015). All degree programs stay with a specific major such as Computer Science, Information Technology, and Information Systems. We could not find any explanations of teaching data analytics for either non-computing majors or underrepresented students.

4. RESEARCH METHOD

EBP in Data Analytics – Workshops

We apply EBP to data analytics education to identify unique values and preferences (Dyba, Kitchenham, & Jorgensen, 2005; Straus, Glasziou, Richardson, Haynes, & Sackett, 2011). As scientific evidences, our literature surveys show that underrepresented students including first-generation-college-bound students need proven and intensive interventions in Science, Technology, Engineering, and Mathematics (STEM) (NAS, 2011). Bettinger and Baker (2014) found that the students who received mentoring kept higher retention.

Based upon NAS recommendations (2011), we propose a series of summer workshops, which are not an elective course for the student participants. Instead of a regular semester, we target a summer semester. We include engagement in networking, peer-to-peer support, study groups, and social activities. Due to the small group setting, we constantly provide them with mentoring.

We also require research experiences, participation in conferences, and presentation of research. Students should apply what they learned in data analytics to their application to experience applied data analytics. In addition, since faculty resources are very limited during summer semester, we educate and train two IT major senior students before summer semester through an independent study. Then, we have the students lead the workshop series as their credits for their second independent study course.

Curriculum – Active Learning Methods

In order to enhance engagement of students, we infuse three active learning methods and the muddiest point into each teaching module. Each teaching module employs the same cycle of active learning method sequence: 1) JiTT with the muddiest point and FC out of classroom, 2) JiTT and PI in classroom, and 3) FC in classroom.

Before starting next class, students were required to study their reading assignment for next workshop. We designed this reading assignment for two purposes: engagement and JiTT. Because the purpose of this assignment was not evaluation but engagement, we allowed unlimited trials for maximum five multiple-choice quizzes. To prepare for taking the quiz, we published lecture-related materials such as links to video clips, articles, and slides, etc. to the Canvas Learning Management System (LMS).

To check whether the students actually conducted their reading assignment and to know what the students could not understand clearly from the given teaching materials, we required a reading assignment quiz. The quiz consists of two types of questions: one for multiple-choice quizzes to check whether the students studied or not before next class, the other for an essay quiz to receive feedback from the students. We designed the last essay question to know the muddiest point of the given reading materials.

Before starting each workshop one day earlier, the instructor (one of two leading students) evaluated the submitted reading assignment and identified what part the students answered correctly or incorrectly from the multiple-choice quizzes and what they really got confused or wanted to learn from the muddiest point question. Then, the instructor adjusted his topics that he will cover in class i.e. the JiTT is implemented. After explaining the confusing or interesting topics, the instructor gave a concept test to the students in class to check whether the students truly understood the key concept of today's class or not. A concept test consists of a quiz with multiple choices. Each student selected an answer and then discussed his or her answer with another classmate who answered differently. Then, the instructor explained the right answer to the students.

After conducting a concept test in the middle of class, the instructor conducted classroom activities for lab assignment. Students practiced R programming with an Integrated Development Environment (IDE), RStudio. Then, they started their labs in class and could continue to finish it out of class. For next class, the instructor requested the students to start the reading assignment for next workshop again after class.

Evidenced Approach – Team Project

At the end of the workshop series, students were required to apply what they learned to a problem domain of their major through a team project. Before starting their team project, the students studied a sample case during weeks 9 and 10. Then, they could apply what they learned to their team project. If necessary, we allowed them to finish the project after week 10.

5. RESULTS

Teaching Modules

We developed the workshop series of data analytics in Spring 2015 with two IT major students. Then, by using the developed workshop series, the IT students conducted a 10-week workshop by tutoring seven non-major students for their second independent study course. We used first nine chapters of Jared Lander's book for R programing with RStudio (Lander, 2013). Table 1 shows the topics of 10 teaching modules. Figure 1 shows that the 10 teaching modules were uploaded onto the Canvas LMS.

Table 1. Teaching Modules for Workshops

W	Teaching Module	Resource
1	Active Learning Methods Getting R	Syllabus Ch. 1
2	R Computing Environment	Ch. 2
3	R Packages	Ch. 3
4	Basic Math Functions	Ch. 4
5	Data Structure	Ch. 5
6	Reading Data & Visualization	Ch. 6 & 7
7	Function Definition & Call	Ch. 8
8	Control Structure	Ch. 9
9	Clustering	Online
10	Classification	Online

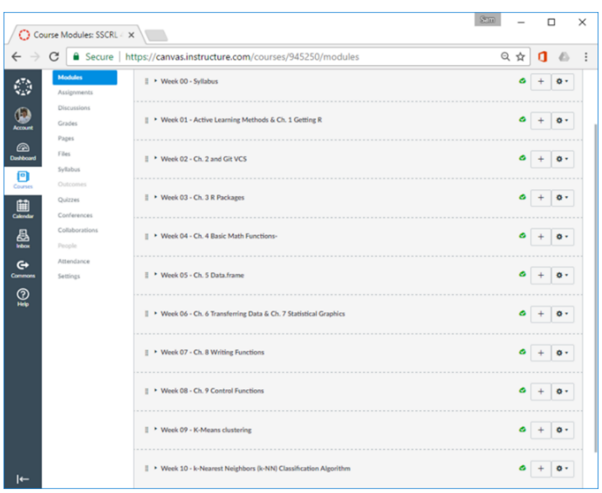

Figure 1: Canvas LMS for Summer Workshop Series

At week 1, we informed students of how the workshop series will be conducted and how active learning methods will be integrated into each teaching module. During next seven weeks, students focused on how to use RStudio and practice R programming. We started with R packages to explain how R can be easily extensible for other business domains. Then, we introduced foundation of R programming - objects, language-defined methods, basic data structures, visualization, user-defined methods, and control structures.

We introduced two well-known k-means clustering and k-nearest neighbor algorithms to the students and practiced the algorithms with Iris Data Set at weeks 9 and 10. For data clustering and classification, we used online resources (Influxity, 2013; Jalayer Academy, 2015a; Jalayer Academy, 2015b) with data samples from University of California Irvine (UCI) Machine Learning Repository (UCI MLR, 2017).

Weekly Workshop

Two IT major students took an independent study course for preparation in spring semester and led the workshop during summer. By using week 9 as an example, we (two students leading the workshop) explain how we conducted a weekly workshop. Table 2 shows which week we took an active learning method and where we exercised the method - out of class or in class.

Table 2. Active Learning Methods (CAT) & Teaching Materials (Where, Where)

Active Learning Methods (CAT)	Teaching Materials (When, Where)
FC (MP)	Course Material for W9 Reading Assignment Quiz for W9 (W8, out of class)
JiTT	Adjusted lectures for W9 (W9, out of class)
PI	Concept Test for W9 (W9, in class)
FC	Lab Assignment for W9 (W9, in class)
FC (MP)	Course Material for W10 Reading Assignment Quiz for W10 (W9, out of class)

FC: Flipped Classroom, MP: the Muddiest Point; JiTT: Just-in-Time Teaching; PI: Peer Instruction

At week 9, the students learned a clustering algorithm with the k-means R package. We held the workshop on Saturday from 9:30 AM to 12:30 PM. There was a reading assignment quiz at the end of week 8 for the students to be engaged in week 9 class. We provided the students with course materials and video clips. We required the students to take a reading assignment quiz including the muddiest point question. Prior to each workshop, we checked the reading assignment quiz to know what concepts we need to explain in next class. After conducting the JiTT, we required the students to take a concept test quiz in class for PI. The students themselves discussed and answered the concept test first and we explained its answer later. Then, the students started hands-on labs in class and could continue the lab after class. We required the students to answer several questions at the end of each lab to confirm whether the students truly understood the lab – why you did this lab. We repeated this cycle throughout 10 weeks.

Three Team Projects

At the end of the workshop series, the students applied what they learned to their application domain trough a team project. Table 3 shows the three-team projects (total 7 non-computing major students). The cases show that the students could apply the Machine Learning approach to their problem domains - identification of automotive vehicles, on-time performance of airline operations, and smoking effect on newborn babies.

Table 3. Applied Data Analytics Projects

C	Majors	#	Title
1	Automotive Technology	2	Vehicle Clustering by Manufacturer Region Using the K-means Clustering Machine Learning Algorithm
2	Aviation Management & Flight	3	K-means Clustering of Airline On-Time Performance Statistics
3	Physiology & Biochemistry	2	Effect of Smoking on Newborn Weight and Length at Birth by Using the K-Means Clustering Algorithm

C: Case; #: the number of participants

In Case 1, the team sought to find how a machine learning approach using the k-means clustering algorithm can be applied to vehicle clustering. The vehicles were clustered into North America, Europe, and Asia regions in terms of engine displacement and Miles Per Gallon (MPG). The team discovered that vehicles from the Asian region were surprisingly clustered quite differently from the actual data set because the vehicles were clustered as European vehicles rather than Asian (Figure 2).

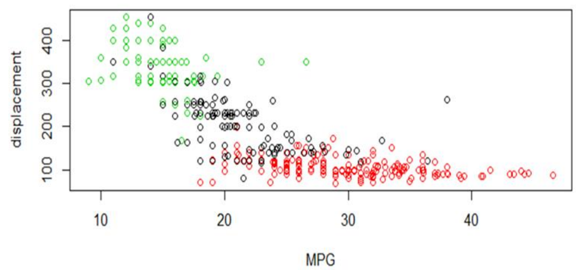

Figure 2. Scatterplot results of clustering engine displacement and MPG according to manufacturer regions (green for North America, black for Europe, and red for Asia) (Chung et al. 2017)

In Case 2, the team analyzed airline on-time performance statistics and found that the lowest clustering group reflecting the low on-time performance rate is mainly associated with winter and summer months (Figure 3).

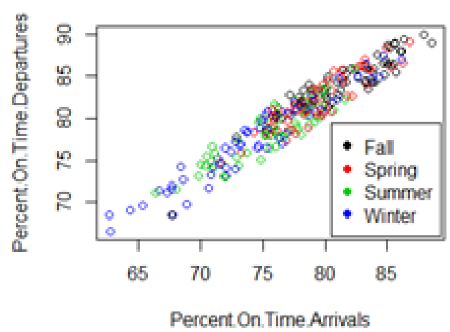

Figure 3. Scatterplot results of clustering flight departures and percentages of on-time arrivals according to four seasons (Chung et al. 2017)

In Case 3, the team analyzed the relationships of the height and weight of infants from non-smoking mothers and mothers with regular contact with cigarette tobacco. The team confirmed that there is a significant difference in the height and weight variables when maternal smoking is a contributing factor (Figure 4). More detail information of three case studies can be found in Chung et al. (2017).

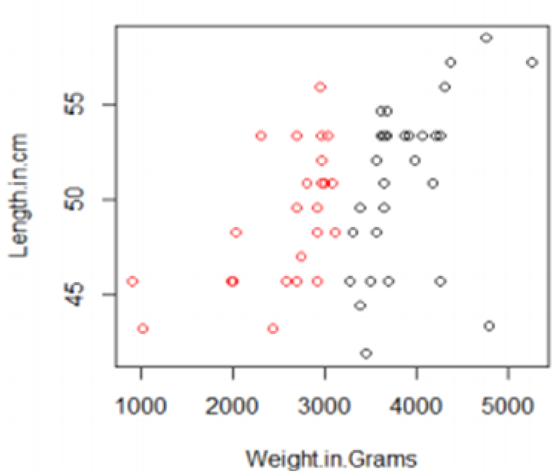

Figure 4. Scatterplot results of clustering length and weight of infants according to maternal smoking (red) and non-smoking (black) (Chung et al. 2017)

6. EVALUATION AND DISCUSSION

Formative Evaluation
Based upon our observation, the percentages of submitting both the reading assignment quiz before next class and the lab assignment quiz after class were not high because of their distraction due to their other priorities. However, they were strongly engaged in the concept test and the labs in class if they did not miss their workshop. Later, we allowed them to finish both the reading assignment and the lab assignment quizzes in class. Then, the percentages increased.

Summative Evaluation
We required each student to be a part of a team project and apply what he or she learned about data analytics to their problem. All students successfully finished their term project. Evan after they finished the workshop series, they worked together since we strongly recommended submission of an abstract to a peer-reviewed student poster session at a local symposium. All three teams participated in submitting their abstract to a university-hosted symposium in Fall 2015. The symposium organizer accepted all of three abstracts after peer review and invited two teams for oral presentation and one team for poster presentation in November 2015 at the symposium.

What We Learned
First, we answer what practices that have underpinning scientific evidences we could use. We chose the summer, student-driven, and applied workshop format for non-computing major first-generation-college-bound students. The proven practices such as active learning for improving student learning, learning by doing for earning skills, team project-based learning for cultural agility and systems thinking, and research experiences for creative thinking were integrated into the teaching modules.

Second, we answer how we could enhance engagement of students with very limited interventions from an instructor. While the intervention from a faculty member was minimized, the interventions among the students were maximized. Two leading students, who were IT major, were educated and trained first through their independent study for one semester (academic learning). Then, they led the workshop series with limited interventions from the instructor for other non-computing major students (experiential learning). They could also learn how the integration of active learning methods into teaching modules could improve student engagement and learning.

Third, we answer how we could motivate the participation of the non-computing major students. The non-computing major students could learn data analytics by applying what they learned to their own data analytics problem. The summer workshop series could support networking, peer-to-peer support, study groups, and social activities. Due to the small group setting, we constantly provided them with mentoring. The inclusion of undergraduate research experiences based upon the outcomes of their team projects helped them to learn how creative thinking, technical writing, and professional presentations are important.

7. CONCLUSION & FUTURE WORK

We propose how we can encourage non-computing major first-generation-college-bound students to be actively involved in learning data analytics. EBP guided us to choose the teaching format, summer workshop, and to implement each teaching module with three active learning methods, one CAT, and one team project. The team project allowed the students to apply data analytics to their own business domain and to experience research and creative activities through a student poster with their abstract. This approach could enhance engagement of the students and motivate the participation of the students.

A series of summer workshops for data analytics that are student-driven brought several benefits to either computing-major students who led the workshops or the non-computing major first-generation-college-bound who participated in the workshops. The leading students had both academic learning through their preparation of the workshops and experiential learning through the workshop series. The non-computing major first-generation-college-bound students could experience hands-on learning and learning by doing through the applied team projects

For next steps, we will expand the workshop series to more diverse groups of students. The current population is limited to male and one ethnic group. Also, we will analyze how these workshop series could help retention, pathways, and graduation of the students participated since we have kept connections with all of nine participant students (2 IT major and 7 non-IT major) since October 2015.

8. REFERENCES

Anderson, A., Bowring, J. McCauley, R., Pothering, G., & Starr, C. (2014). An undergraduate degree in data science: curriculum and a decade of implementation experience, *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14)*. Atlanta, GA. March 5-8, 2014. 145-150.

Angelo, T. A., & Cross, K. P. (1993). Classroom assessment techniques: A handbook for college teachers. *San Francisco: Jossey-Bass*.

Bergmann, J., & Sams, A. (2012). *Flip Your Classroom. Reach Every Student in Every Class Every Day.* Washington, DC: International Society for Technology in Education.

Bettinger, E.P., & Baker, R. (2014). The Effects of Student Coaching: An Evaluation of a Randomized Experiment in Student Advising. *Educational Evaluation and Policy Analysis*, 36(1), 3-19.

Chung, S., Oh, T. J., Kim, J. Kang. A. (2017). Data Analytics Workshop with First-Generation-College-Bound, Underrepresented, and Non-Computing Major Students. *Consortium for Computing Sciences in Colleges Midwest Conference (CCSC MW 2017)*, September 22-23, 2017. Grand Rapids, MI.

Dyba, T., Kitchenham, B. A., & Jorgensen, M. (2005). Evidence-based software engineering for practitioners. *IEEE software*, *22*(1), 58-65.

Gurka. J. S. (2012). JiTT IN CS 1 AND CS 2. *Journal of Computing Sciences in Colleges*, Volume 28, Issue 2, December 2012, p. 81-86.

Influxity. (2013) How to Perform K-Means Clustering in R Statistical Computing, Retrieved June 15, 2017. https://www.youtube.com/watch?v=sAtnX3UJyN0

Jafar, M. J., Babb, J., & Abdullat, A. (2016). Emergence of Data Analytics in the Information Systems Curriculum. In *Proceedings of the EDSIG Conference ISSN* (Vol. 2473, p. 3857).

Jalayer Academy. (2015a). R - kNN - k nearest neighbor (part 1), Retrieved June 15, 2017. https://www.youtube.com/watch?v=GtgJEVxl7DY

Jalayer Academy. (2015b). R - kNN - k nearest neighbor (part 2), Retrieved June 15, 2017. https://www.youtube.com/watch?v=DkLNb0CXw84

Kang, J. W., Holden, E. P., & Yu, Q. (2014, October). Design of an analytic centric MS degree in information sciences and technologies. In *Proceedings of the 15th Annual Conference on Information technology education* (pp. 147-152). ACM.

Kang, J. W., Holden, E. P., & Yu, Q. (2015, September). Pillars of Analytics Applied in MS Degree in Information Sciences and Technologies. In *Proceedings of the 16th Annual Conference on Information Technology Education* (pp. 83-88). ACM.

Kitchenham, B. A., Dyba, T., & Jorgensen, M. (2004, May). Evidence-based software engineering. In *Proceedings of the 26th international conference on software engineering* (pp. 273-281). IEEE Computer Society.

Kumar, S. A. (2014, October). Designing a graduate program in information security and analytics: master's program in information security and analytics (MISA). In *Proceedings of the 15th Annual Conference on Information technology education* (pp. 141-146). ACM.

Lander, J. (2014). R for Everyone: Advanced Analytics and Graphics. Upper Saddle River, New Jersey: Addison-Wesley.

Martinez, A. (2012). Using JiTT in a Database Course, the Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE'12), p. 367-372.

Mazur, E. (1997). Peer Instruction: A User's Manual. Prentice Hall.

Mazur, E. & Watkins, J. (2010). Just-in-Time Teaching and Peer Instruction. Just-in-Time Teaching, edited by Simkins, S. and Maier, M., Stylus, Publishing, LLC, Sterling, VA.

National Academic of Sciences (NAS). (2011). Expanding Underrepresented Minority Participation. The National Academic Press. Washington, D.C.

Novak, G., Patterson, E., Gavrin, A., & Christian, W. (1999). *Just-in-Time Teaching: Blending Active Learning and Web Technology*. Prentice Hall.

Rutherfoord, R. H., & Rutherfoord, J. K. (2013, October). Flipping the classroom: Is it for you? In *Proceedings of the 14th annual ACM SIGITE conference on Information technology education* (pp. 19-22). ACM.

Sackett, D. L., Rosenberg, W., Gray, J. A. M., Haynes, R. B., & Richardson, W. S. (1996). Evidence based medicine: what it is and what it isn't. *BMJ* 312, 71072.

Simon B., & Cutts, Q. (2012). How to implement a peer instruction designed CS principles course. *ACM Inroads*, Volume 3, Issue 2, June 2012. pp. 72-74.

Straus S. E., Glasziou, P., Richardson, W. S., Haynes, R. B., & Sackett, D. (2011). Evidence-Based Medicine: How to Practice and Teach It, 4th edition. Churchill Livingstone, Edinburgh, 2011, p.1

The Executive Office of the President. (2014). Increasing College Opportunity for Low-Income Students. The White House. January 2014.

UCI MLR. (2017). The UC Irvine Machine Learning Repository, Retrieved March 17, 2017. http://archive.ics.uci.edu/ml/

Wright, A. A. (2016). Jobs of the Future Will Require Data Analysis. *Society for Human Resource Management*. Retrieved March 17, 2017 from https://www.shrm.org/resourcesandtools/hr-topics/technology/pages/jobs-of-the-future-will-require-data-analysis.aspx

Wymbs, C. (2016). Managing the Innovation Process: Infusing Data Analytics into the Undergraduate Business Curriculum (Lessons Learned and Next Steps). *Journal of Information Systems Education,* 27(1), 61-74.

Long-term Follow-up of STEM Scholarship Students to Degree Attainment

Sylvia Sorkin
ssorkin@ccbcmd.edu
Mathematics Department
The Community College of Baltimore County
Baltimore, MD 21237 USA

Abstract

This paper describes the results of long-term follow-up of need-based scholarship awardees at a community college as they made progress toward their goal of associate's degrees and/or bachelor's degrees in Science, Technology, Engineering, and Mathematics (STEM) fields. From 2004 through 2012, through National Science Foundation funding, need-based scholarships were offered for full-time STEM students with a minimum grade point average, and U.S. citizenship, or status as permanent resident alien or refugee alien. Faculty mentoring, a seminar luncheon series, and career information were used to increase degree attainment or transfer in STEM fields. Outcomes of these efforts are described, including time elapsed from initial enrollment in the community college to subsequent bachelor's degree attainment. Outcomes by gender, race/ethnicity, and initial mathematics placement of awardees are also provided.

Keywords: time to degree, mentoring, scholarships, transfer rate, underrepresented, STEM

1. INTRODUCTION

The STEM Executive Summary noted that with respect to incomes, "People with an undergraduate major in STEM make substantially more over their lifetimes than non-STEM majors," (Carnevale, Smith, Melton, 2011). Nonetheless, women's bachelor's degree attainment in science and engineering declined in every field from 2004 to 2014. In 2014, women earned 19% of engineering and 18% of computer science bachelor's degrees (Espinosa, 2015). Underrepresented minorities in STEM include African Americans and Hispanic/Latinos. Although African Americans make up 13.2% of the U.S. population, they represent only 4% of engineering bachelor's degree recipients. Similarly, Latinos comprise 17.5% of the U.S. population, but represent only 9% of engineering bachelor's degrees (Chang, 2015).

The National Science Foundation's Division of Undergraduate Education provides a program known as Scholarships for Science, Technology, Engineering, and Mathematics (S-STEM) that can address the underrepresentation described above (NSF, 2017). This program makes grants to institutions of higher education, which in turn are responsible for selecting scholarship recipients and reporting demographic information about student scholars. Scholarship recipients must be academically talented but financially needy and enrolled full-time in one of these programs: computer and information sciences, engineering, mathematical sciences, biological sciences, physical sciences, geosciences, or technology areas associated with those fields. Individual scholarships cannot exceed $10,000 per year.

The S-STEM program was preceded by NSF's similar but more restrictive Computer Science, Engineering, and Mathematics Scholarship (CSEMS) program which provided funds to institutions of higher education to select full-time financially needy scholarship recipients from these degree programs: computer science, computer technology, engineering, engineering technology, or mathematics. CSEMS scholarships could not exceed $3,125 per year.

For both the CSEMS and S-STEM programs, the individual college/university determines award criteria, including minimum GPA and eligible major programs. However, NSF guidelines specify that students who are awarded these scholarships must be U.S. citizens, permanent residents, nationals, or refugees.

This paper describes long-range degree outcomes for two specific CSEMS and S-STEM scholarship programs at the Community College of Baltimore County. The scope of the paper includes demographics of the combined CCBC awardees, and transfer and graduation rates, for all awardees and certain subgroups of awardees, including by gender, race and ethnicity. Awardees' time elapsed from initial enrollment at this community college to bachelor's degree attainment is also presented.

2. BACKGROUND INFORMATION

The *Time to Degree* research report (Shapiro, Dundar, Wakhungu, Yuan, Nathan & Hwang, 2016) supported by the Lumina Foundation measured time to degree in 2 different ways: elapsed time to degree, and enrolled time to degree. The first measurement (and the one used in this paper) was the time that elapsed between students' first term begin date and the date of degree award. Time elapsed was defined as, "the total time, in calendar years, between initial enrollment in a postsecondary institution and subsequent degree attainment, regardless of whether or not the student was actually enrolled."

In that study, the average elapsed time was 5.7 years for bachelor's degree earners. Those authors found that the average time elapsed to bachelor's degree was extended for students who earned an associate's degree prior to receiving their bachelor's. For students with a prior associate's degree, the time elapsed to a bachelor's degree was 8.2 years. For students without an associate's degree, the time elapsed to bachelor's degree was 5.1 years. Bachelor's degree earners without a prior associate's degree may or may not have also attended a two-year institution. Among bachelor's degree earners with no associate's degree, the average time elapsed to bachelor's degree was 6.0 years for those with prior enrollments in 2-year institutions, and was 4.5 years for those without prior enrollments in 2-year institutions.

The second measurement (called enrolled time, which is not used in this paper) was the actual time in academic years that the student was enrolled full-time (or its full-time equivalent) in postsecondary institutions. (Shapiro et al., 2016)

3. INSTITUTIONAL INFORMATION

The Community College of Baltimore County (CCBC) is a public two-year college system with three campuses serving the greater Baltimore metropolitan area. The Fall 2015 combined credit enrollment was 22,179 students of which 29% were full-time students. Thirty-nine percent (39%) of the credit students were African American, and 60% of credit students were female. The institutional rate of Pell awards provides one indication of the level of unmet financial need. In the 2013-14 academic year, 45% of the credit students at CCBC received a Pell grant. In FY 2015, CCBC awarded 2,200 associate's degrees.

From 2010 to 2015, although CCBC's total fall enrollment steadily declined (falling 16% over that period), enrollment in STEM associate's degree programs increased 43%. Within STEM associate's degree programs at CCBC, the largest enrollment increases occurred in Network Technology (102%), Computer Science (39%), and the new Information Systems Security program which began in 2011. Over that period, there was a 152% increase in the number of STEM associate's degrees awarded at CCBC (MHEC, June 2016a, Nov. 2016). These increases are shown in **Figure 1** (see Appendix).

4. CSEMS AND S-STEM SCHOLARSHIP PROGRAMS AT CCBC

Two Specific Scholarship Programs

One of the CCBC scholarship programs was a CSEMS program, which awarded renewable semester scholarships to 75 (25 female and 50 male) full-time students from Fall 2004 through Fall 2008 (Sorkin, Gore, Mento & Stanton, 2010). The other scholarship program was an S-STEM program, which awarded renewable semester scholarships to 99 (36 female and 63 male) full-time students from Fall 2008 through Fall 2012 (Sorkin, 2013).

CSEMS Project at CCBC

NSF funding for the four-year CSEMS scholarship project, *Promoting Computer Science, Engineering, and Mathematics with Scholarships and Student Support Services,* (DUE-0422225), enabled CCBC to award semester scholarships to a total of 75 students. Award criteria included a minimum 2.5 GPA and readiness for MATH 082 Introductory Algebra, or higher. Computing field associate's degree programs targeted by this project were: the Multimedia Technology (MULT) program that includes the 2+2 Simulation and Digital Entertainment bachelor's degree program with a local four-year university; the Computer Science (CMSC) transfer program; and the Computer Information Systems (CINS) and Data Communications and Network Technology (DCOM) programs that prepare students for entry-level employment or transfer to a four-year institution. The E-Business (EBUS) career program was also targeted, along with the Mathematics (MATH) and Engineering (ENGR) transfer programs.

S-STEM Project at CCBC
The NSF-funded four-year project at CCBC, *STEM Scholars Community*, provided renewable scholarships of up to $10,000 (but not to exceed unmet financial need as determined by FAFSA) per year for full-time students with minimum 2.8 GPA majoring in one of these 7 transfer programs: Biology (BIOL), Chemistry (CHEM), Computer Science (CMSC), Engineering (ENGR), Environmental Science (ENVS), Mathematics (MATH), and Physics (PHYS). Award criteria included readiness for MATH 083 Intermediate Algebra, or higher. A total of 99 students received semester scholarship awards under this program. Awardees were also required to take a MATH course each semester until all mathematics required for their major program was completed. Nine (9) students initially received CSEMS and later received S-STEM scholarships. Duplicate counts are removed from the combined list of awardees shown in **Figure 2**.

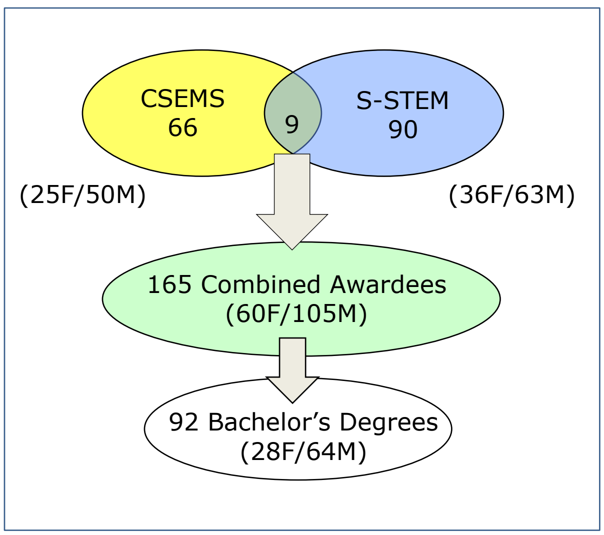

Figure 2. CCBC CSEMS and S-STEM Awardees Combined from Fall 2004 through Fall 2012.

Efforts to Increase the Transfer Rate
Both of these scholarship programs encouraged awardees to continue their STEM studies at four-year institutions. Each project designated a portion of its scholarship funds to "follow" awardees who transferred and to thereby assist awardees to complete bachelor's degrees in these fields.

Transferring awardees had to provide documentation of their: acceptance and full-time status at the four-year institution in a STEM major; unmet financial need; and successful completion of prior coursework in a STEM degree program at CCBC. Students were given the option of transferring their CSEMS or S-STEM scholarship along with their credits to a four-year institution. This enabled students to reach greater success in a STEM field by drawing attention to the possibility of transfer to a four-year school for students who otherwise might not have considered transfer.

Transfers among Awardees
Thirty-four percent (34%) of the CSEMS semester scholarship awards were made as scholarship renewals to awardees who had earned at least 30 credits at CCBC and were transferring to a 4-year institution. For the S-STEM program, 27% of semester scholarship awards were made to former awardees who had earned an associate's degree, or at least 45 credits, at CCBC before transferring.

5. AWARDEE OUTCOMES

Eighty-seven percent (87%) of the 165 total awardees transferred to a 4-year institution, and an additional 6% earned associate's degrees but did not transfer to a 4-year institution. As shown in **Figure 3**, 55% of the 165 combined awardees earned associate's degrees, 56% earned bachelor's degrees, and 4% earned Doctor of Pharmacy degrees (as of June 2016). And 84% earned at least one of these degrees.

S-STEM Awardee Outcomes by Initial Mathematics Placement Level at CCBC
Students entering CCBC take an *Accuplacer* mathematics placement test which determines their initial mathematics placement. Over half of entering students have an initial mathematics placement that is developmental. That is, the student places into one of the 3 non-credit developmental mathematics courses: MATH 081 Pre-Algebra, MATH 082 Introductory Algebra, or MATH 083 Intermediate Algebra.

S-STEM awardees at CCBC were required to have eligibility for MATH 083 Intermediate Algebra, or higher, at the time of their award. They could obtain this eligibility through their initial mathematics placement, or by completing any needed developmental mathematics courses. Among the total 99 (36F/63M) CCBC S-STEM awardees from Fall 2008 through Fall 2012 were 37 whose initial mathematics placement level at CCBC was developmental. Although most (30) of these 37 awardees initially placed into MATH 083, there were 4 who initially placed into MATH 082, and 3 who initially placed into MATH 081. The outcomes and success rates for the 37 awardees with initial placement into developmental mathematics, and for the other 62 awardees with initial placement into non-developmental mathematics are shown in **Figure 4**. Among those who initially placed into developmental mathematics, 89% (33/37) transferred to a 4-year institution or graduated with an associate's degree (but did not transfer). Among those whose initial placement was into non-developmental mathematics, 97% (60/62) transferred to a 4-year institution or graduated with an associate's degree. Among the 99 total S-STEM awardees, females formed 35% (13/37) of the developmental initial placement group, and 37% (23/62) of the non-developmental initial mathematics placement group of 99 total awardees.

Awardee Outcomes by Gender

Considering all 165 combined awardees from Fall 2004 through Fall 2012, as of June 2016, a total of 144 awardees (87%) have transferred to 4-year colleges/universities. Overall, 87% (52/60) of the female awardees, and 88% (92/105) of the male awardees have transferred. Bachelor's degrees were earned by 92 awardees (92/165 = 56%), including 47% of female awardees and 61% of male awardees. This is shown in **Figure 5.**

The institutions from which most awardees earned their bachelor's degrees were: 39 from the University of Maryland Baltimore County (UMBC), 11 from Towson University (TU), and 10 from the University of Maryland College Park (UMCP), as well as several other 4-year institutions (most of them public and in-state). Major programs for those 92 who have earned bachelor's degrees are: ENGR (29), IS/IT/Networks (15), BIOL (8), MATH (8), CMSC (5), CHEM (4), PHYS (3), Multimedia (3), and non-STEM programs (17). This is shown in **Figure 6**.

Demographics of CSEMS and S-STEM Awardees

The distribution of all credit students at CCBC in Fall 2012 by racial/ethnic group as self-described at course registration was as follows: white 47%, African American 38%, Asian 5%, Hispanic/Latino 4%, and Other/Unknown 6% (MHEC, June 2016b). Minority groups that have been under-represented in STEM fields nationally are represented among the 165 CSEMS and S-STEM awardees (from Fall 2004 through Fall 2012) in proportions close to their population percentage at CCBC. In particular, 44% of the 165 awardees were white, 30% were African American, 19% were Asian, 4% were Hispanic/Latino, and 2% were Other/Unknown, as shown in **Figure 7**.

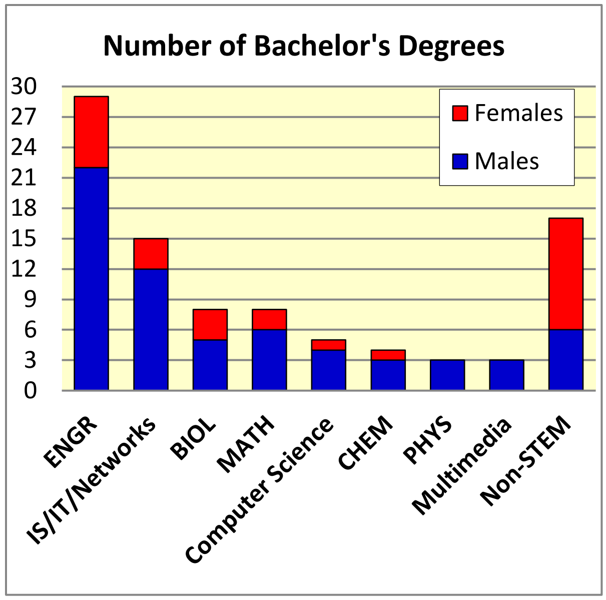

Figure 6. Major Programs for Earned Bachelor's Degrees for 92 CCBC Combined CSEMS and S-STEM Awardees from Fall 2004 through Fall 2012 by Gender.

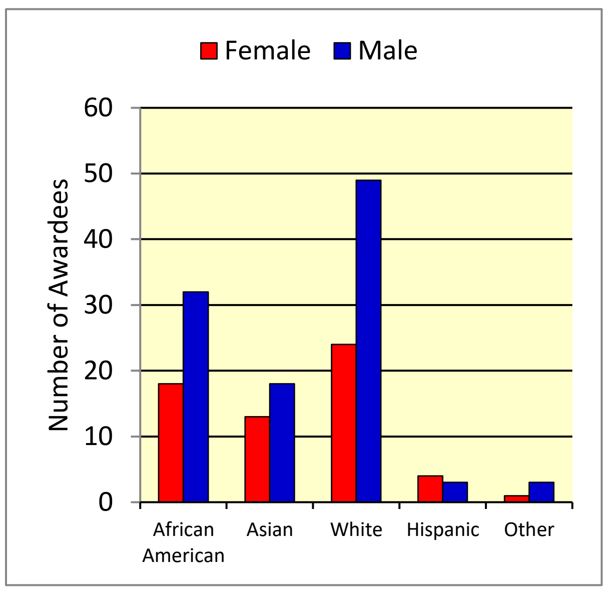

Figure 7. CCBC's 165 CSEMS and S-STEM Awardees from Fall 2004 through Fall 2012 by Race/Ethnicity and Gender.

Awardee Outcomes by Racial/Ethnic Group
As shown in **Figure 8**, although 87% of the 165 combined awardees transferred to a 4-year institution, the transfer rate was higher (94%) among African American awardees, and lower (78%) among white awardees. Although 55% of all awardees earned associate's degrees, 62% of African American awardees, and 62% of white awardees earned associate's degrees. This is shown in **Figure 9**.

Time Elapsed to Bachelor's Degree
The time elapsed to bachelor's degree was determined from awardees' initial entry to CCBC (in months). As shown in **Figure 10**, the median time elapsed was 63 months (5.3 years). The mean time elapsed was 71 months (5.9 years). The distribution of time elapsed to bachelor's degree was skewed to the right. The mean time was affected by awardees who took up to 180 months (15 years) to earn their bachelor's degree, taking time out from coursework at the 4-year institution while working full-time.

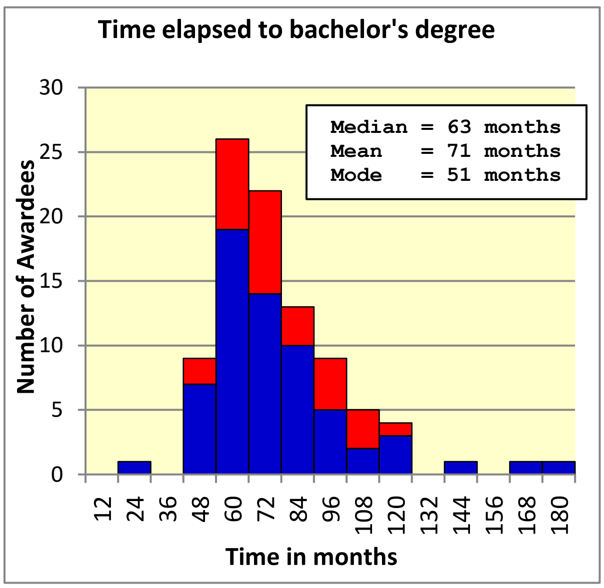

Figure 10. Time Elapsed to Bachelor's Degree from CCBC Entry for 92 CSEMS and S-STEM Awardees from Fall 2004 through Fall 2012 by Gender.

6. CONCLUSIONS

From Fall 2004 through Fall 2012, 165 full-time CCBC students majoring in certain STEM fields received CSEMS or S-STEM scholarships for one or more semesters through NSF funding.

These awardees have been highly successful in graduating with associate's degrees and/or transferring to 4-year institutions where 92 have earned bachelor's degrees.

The transfer rate to 4-year institutions was higher among African American awardees (94% transferred), than among white awardees (78% transferred). African American awardees also had a higher rate (56%) of earning bachelor's degrees than white awardees (49%). Associate's degree attainment was equal (62%) for African American and white awardees.

Awardees with initial developmental and non-developmental mathematics placements were equally likely (49% and 53%) to earn associate's degrees.

Female (87% transferred) and male awardees (88% transferred) were equally likely to transfer to 4-year institutions. Female awardees were 30% more likely to earn associate's degrees than male awardees. Male awardees were 30% more likely to earn bachelor's degrees than female awardees.

Based on these results, for community colleges trying to increase their percentage of students who earn associate's degrees, it seems advisable to focus increased efforts on male students in STEM fields for associate's degree completion. For four-year institutions trying to increase their percentage of transfer students who earn bachelor's degrees in STEM fields, it may be advisable to focus additional efforts on female transfer students for bachelor's degree completion.

Twenty-four percent (24%) of bachelor's degrees earned by awardees were in IS/IT/Networks, Computer Science, and Multimedia major programs combined. Thirty-two percent (32%) of earned bachelor's degrees were in Engineering programs, and 19% were in non-STEM major programs.

Awardees' mean time elapsed to bachelor's degree was 5.9 years, and the median time elapsed to bachelor's degree was 5.3 years, for this group of scholarship awardees.

7. ACKNOWLEDGEMENTS

This material is based upon work supported in part by the National Science Foundation under awards DUE-0422225 and DUE-0806664. Opinions expressed are those of the author and do not necessarily reflect the views of the NSF.

8. REFERENCES

Carnevale, A., Smith, N. & Melton, M. (2011, Oct. 20). *STEM Executive Summary.* Georgetown Center on Education and the Workforce. https://cew.georgetown.edu/wp-content/uploads/2014/11/stem-execsum.pdf

Chang, J. (2015, Nov. 8). Bridging the Racial Gap in STEM Education. NACME. http://nacme.org/news/articles/170-bridging-the-racial-gap-9n-stem-education

Espinosa, L. (2015, March 3). Where Are the Women in STEM?. *Higher Education Today*. https://www.higheredtoday.org/2015/03/03/where-are-the-women-in-stem/

MHEC Maryland Higher Education Commission. (Nov. 2016). *Trends in Degrees and Certificates by Program: Maryland Higher Education Institutions (2003-2016).* http://mhec.maryland.gov/publications/Documents/Research/AnnualReports/Degrees2016ByProgram.pdf.pdf

MHEC Maryland Higher Education Commission. (June 2016a). *Trends in Enrollment by Program: Maryland Higher Education Institutions (2002-2015).* http://mhec.maryland.gov/publications/Documents/Research/AnnualReports/2015EnrollmentbyProgram.pdf

MHEC Maryland Higher Education Commission. (June 2016b). *Trends in Enrollment by Race and Gender: Maryland Higher Education Institutions (2006-2015).* http://mhec.maryland.gov/publications/Documents/Research/AnnualReports/2015EnrollmentbyRaceandGender.pdf

NSF. (2004). NSF Computer Science, Engineering, and Mathematics Scholarships (CSEMS). https://www.nsf.gov/pubs/2004/nsf04506/nsf04506.htm

NSF. (2017). NSF Scholarships in Science, Technology, Engineering, and Mathematics Program (S-STEM). https://www.nsf.gov/pubs/2017/nsf17527/nsf17527.htm

Shapiro, D., Dundar, A., Wakhungu, P.K., Yuan, X., Nathan, A. & Hwang, Y. (2016, September). *Time to Degree: A National View of the Time Enrolled and Elapsed for Associate and Bachelor's Degree Earners* (Signature Report No. 11). Herndon, VA: National Student Clearinghouse Research Center. https://nscresearchcenter.org/signaturereport11/

Sorkin, S. (2013). Increasing Graduation and Transfer Rates for STEM Students in the Community College. *2013 Proceedings of the Information Systems Educators Conference,* v30 n2538. ISSN: 2167-1435.

Sorkin, S., Gore, M., Mento, B., & Stanton, J. (2010). Tracking Women and Minorities as They Attain Degrees in Computing and Related Fields. *Information Systems Education Journal, 8* (50). http://isedj.org/8/50/. ISSN: 1545-679X.

.

Appendices

CCBC Associate's Degree Program	Fall Enrollment in Program Major						Associate Degrees Awarded					
	2010	2011	2012	2013	2014	2015	FY 2010	FY 2011	FY 2012	FY 2013	FY 2014	FY 2015
* Science	635	583	551	624	724	746	21	22	27	33	37	44
Engineering	353	391	494	464	413	406	3	7	13	13	9	20
Computer Engineering	0	0	0	0	9	56	0	0	0	0	0	0
Electrical Engineering	0	0	0	0	7	45	0	0	0	0	0	1
Computer Science	340	423	454	415	442	473	7	9	15	14	28	21
Secondary Ed - Chemistry	8	7	10	9	6	2	0	0	0	0	0	0
Secondary Ed - Mathematics	25	33	26	37	30	26	1	0	2	2	0	1
Secondary Ed - Physics	7	3	4	5	4	4	0	0	0	0	0	0
Geospatial Applications	28	25	31	30	29	42	4	3	1	7	9	1
Information Technology	292	306	257	308	283	304	12	21	26	23	39	33
Information Systems Security	0	16	81	140	139	233	0	0	4	8	32	29
Network Technology	191	200	242	435	438	385	27	35	37	34	47	34
Engineering Technology	101	120	111	140	129	117	0	2	5	8	5	5
Totals:	1,980	2,107	2,261	2,607	2,653	2,839	75	99	130	142	206	189
CCBC Total Fall Enrollment and Total Associate's Degrees	26,425	26,271	25,188	24,275	22,887	22,179	1,703	1,854	2,132	2,086	2,020	2,200
%STEM Enrollment and Degrees	7%	8%	9%	11%	12%	13%	4%	5%	6%	7%	10%	9%

*Biology, Chemistry, Environmental Science, Mathematics, and Physics data are included in the Science program.

Sources: MHEC Trends in Enrollment by Program (June 2016), MHEC Trends in Degrees and Certificates by Program (March 2016), MHEC Opening Fall Enrollment (November 2011, December 2012, November 2013, November 2014, November 2015), and MHEC Data Book 2016, 2015, 2014, 2013, 2012, 2011.

Figure 1. CCBC Enrollment and Associate's Degrees Awarded in STEM Programs, 2010 – 2015.

Awardee Outcome	CSEMS Fall 04 – Fall 08		S-STEM Fall 08 – Fall 12		Combined Fall 04 – Fall 12	
	Number of Awardees	**% of Awardees**	**Number of Awardees**	**% of Awardees**	**Number of Awardees**	**% of Awardees**
Transferred to 4-yr	65 (21F/44M)	87%	88 (32F/56M)	89%	144 (52F/92M)	87%
Graduated (Associate's degree but no transfer)	5 (2F/3M)	7%	5 (2F/3M)	5%	10 (4F/6M)	6%
Still Enrolled (in community college in 2015)	1 (1F/0M)	1%	1 (0F/1M)	1%	2 (1F/1M)	1%
Dropped Out	4 (1F/3M)	5%	5 (2F/3M)	5%	9 (3F/6M)	5%
TOTALS:	**75 (25F/50M)**	**100%**	**99 (36F/63M)**	**100%**	**165 (60F/105M)**	**100%**
Associate's Degrees:	42 (18F/24M)	56%	51 (22F/29M)	52%	91 (39F/52M)	55%
Bachelor's Degrees:	44 (10F/34M)	59%	57 (19F/38M)	58%	92 (28F/64M)	56%
PharmD Degrees:			6 (4F/2M)	6%	6 (4F/2M)	4%
Earned at least one of these degrees (Associate's, Bachelor's or PharmD):	65 (21F/44M)	87%	82 (30F/52M)	83%	138 (50F/88M)	84%

*Note: 9 students (1F/8M) received CSEMS and subsequently S-STEM awards. They all Transferred and earned Bachelor's degrees. Two of these students (1F/1M) also earned Associate's degrees. Entries in the Combined column eliminate this duplication.

Figure 3. CCBC CSEMS and S-STEM Awardees and Outcomes (as of 6/6/16) from Fall 2004 through Fall 2012.

	Developmental MATH Level		Non-Developmental MATH Level		Combined	
Awardee Outcome	**Number of Awardees**	**% of Awardees**	**Number of Awardees**	**% of Awardees**	**Number of Awardees**	**% of Awardees**
Transferred to 4-yr	30	81%	58	94%	88	89%
Graduated (Associate's degree but no transfer)	3	8%	2	3%	5	5%
Still Enrolled (in community college in 2015)	1	3%	0	0%	1	1%
Dropped Out	3	8%	2	3%	5	5%
TOTALS:	37 (13F/24M)	100%	62 (23F/39M)	100%	99 (36F/63M)	100%
Associate's Degrees:	18	49%	33	53%	51	52%
Bachelor's Degrees:	19	51%	38	61%	57	58%
PharmD Degrees:	2	5%	4	6%	6	6%

Figure 4. Awardee Outcomes for 99 S-STEM Scholars by Initial Mathematics Placement Level upon Entry to CCBC.

	Females		Males		Combined	
Awardee Outcome	**Number of Awardees**	**% of Awardees**	**Number of Awardees**	**% of Awardees**	**Number of Awardees**	**% of Awardees**
Associate's Degrees:	39	65%	52	50%	91	55%
Transferred to 4-yr	52	87%	92	88%	144	87%
Bachelor's Degrees:	28	47%	64	61%	92	56%
PharmD Degrees:	4	7%	2	2%	6	4%
TOTALS:	60		105		165	

Figure 5. Awardee Outcomes for 165 CCBC CSEMS and S-STEM Scholars by Gender.

Awardee Outcome	Combined Fall 04 – Fall 12 Awardees		Racial/Ethnic Group									
			White		African American		Asian		Hispanic/Latino		Other	
	Number of Awardees	% of Award-ees	Number of Awardees	% of Award-ees	Number of Awardees	% of Award-ees	Number of Awardees	% of Award-ees	Number of Awardees	% of Award-ees	Number of Awardees	% of Award-ees
Transferred to 4-yr	144 (52F/92M)	87%	57 (17F/40M)	78%	47 (17F/30M)	94%	31 (13F/18M)	100%	6 (4F/2M)	86%	3 (1F/2M)	75%
Graduated (Associate's degree but no transfer)	10 (4F/6M)	6%	9 (4F/5M)	12%	1 (0F/1M)	2%						
Still Enrolled (in community college in 2015)	2 (1F/1M)	1%			2 (1F/1M)	4%						
Dropped Out	9 (3F/6M)	5%	7 (3F/4M)	10%					1 (0F/1M)	14%	1 (0F/1M)	25%
TOTALS:	**165 (60F/105M)**	**100%**	**73 (24F/49M)**	**100%**	**50 (18F/32M)**	**100%**	**31 (13F/18M)**	**100%**	**7 (4F/3M)**	**100%**	**4 (1F/3M)**	**100%**
% Female in that Racial/Ethnic Group:	**60/165**	**36%**	**24/73**	**33%**	**18/50**	**36%**	**13/31**	**42%**	**4/7**	**57%**	**1/4**	**25%**

Figure 8. Awardee Outcomes for 165 CCBC CSEMS and S-STEM Scholars by Racial/Ethnic Group.

Awardee Degree Outcome	Combined Fall 04 – Fall 12 Awardees		Racial/Ethnic Group									
			White		African American		Asian		Hispanic/Latino		Other	
	Number of Awardees	% of Award-ees	Number of Awardees	% of Award-ees	Number of Awardees	% of Award-ees	Number of Awardees	% of Award-ees	Number of Awardees	% of Award-ees	Number of Awardees	% of Award-ees
Associate's Degrees:	91 (39F/52M)	55%	45 (16F/29M)	62%	31 (14F/17M)	62%	11 (7F/4M)	35%	3 (2F/1M)	43%	1 (0F/1M)	25%
Bachelor's Degrees:	92 (28F/64M)	56%	36 (8F/28M)	49%	28 (9F/19M)	56%	22 (7F/15M)	71%	5 (3F/2M)	71%	1 (1F/0M)	25%
PharmD Degrees:	6 (4F/2M)	4%	1 (0F/1M)	1%	3 (2F/1M)	6%	2 (1F/1M)	6%				
Earned at least one of these degrees (Associate's, Bachelor's or PharmD):	138 (50F/88M)	84%	62 (20F/42M)	85%	43 (16F/27M)	86%	26 (10F/16M)	84%	5 (3F/2M)	71%	2 (1F/1M)	50%
TOTALS:	165 (60F/105M)	100%	73 (24F/49M)	44%	50 (18F/32M)	30%	31 (13F/18M)	19%	7 (4F/3M)	4%	4 (1F/3M)	2%

Figure 9. Degree Outcomes for 165 CCBC CSEMS and S-STEM Scholars by Racial/Ethnic Group

Do the Knowledge and Skills Required By Employers of Recent Graduates of Undergraduate Information Systems Programs Match the Current ACM/AIS Information Systems Curriculum Guidelines?

Timothy Burns
tburns1@ramapo.edu

Yuan Gao
ygao@ramapo.edu

Cherie Sherman
csherman@ramapo.edu

Stephen Klein
sklein@ramapo.edu

Anisfield School of Business
Ramapo College of New Jersey
Mahwah, NJ 07430 USA

Abstract

This research investigates the knowledge and skills required by potential employers of students graduating from undergraduate Information Systems programs. For this study, job listings were collected and analyzed from several Internet web sites specializing in technology related employment. The job listings, collected over a four month period, were for entry level jobs that specifically required an undergraduate degree in Information Systems or a related program. The results show that potential employers are most interested in "soft skills" such as written and oral communication, teamwork, and problem solving skills as well as general technical skills. The article then compares the knowledge and skills required by potential employers to the suggested curriculum of the 2010 ACM/AIS Information Systems Curriculum Guidelines.

Keywords: Information Systems Knowledge and Skills, Information Systems Curriculum

1. INTRODUCTION

Current industry demand for employees with technology skills is well documented (Burns, Gao, Sherman, & Klein 2014). Increasingly, many of those employees in demand, especially at the entry level, are recent graduates of collegiate undergraduate Information Systems (IS) programs. Accordingly, educators in the IS field want to make sure that their programs meet the requirements of their stakeholders (Pierson, Kruck, & Teer, 2009). As the significance of information systems in the business world increases, so does the importance of the IS curriculum (Tehrani, 2015). Additionally, while there has been some improvement in recent years, enrollment in the college majors that would prepare students for careers in technology oriented jobs is significantly less than it was at the turn of the century (Burns, Gao, Sherman, & Vengerov, 2014). One suggestion for improving enrollment is to change the IS curriculum (Tehrani, 2015).

This research has multiple goals. The first goal is to gain a general understanding of the current knowledge and skills that are most in demand by employers of the students graduating from IS programs. The second goal is to compare those skills to the most recent model IS curriculum that is used by many IS programs in colleges and universities around the world. Finally, the ultimate goal is to provide insight into a revised IS model curriculum that would prepare students to have the knowledge and skills that are highest in demand by employers.

2. LITERATURE REVIEW

At least as far back as the 1980s, IS faculty reviewed the content of job advertisements to better understand the skills graduating IS students would require in order to be successful in the workforce (Knodel, 1982). Due to the popularity of online recruiting websites, there is every reason to believe that job advertisements continue to be essential to recruiting and it is estimated that about three-fourths of IS jobs are advertised (Litecky, 2012). Litecky (2012) provide a useful summary of the many job advertisements studies, the majority of which concluded that employers were looking for so-called "hard skills" or technical skills such as programming and database management.

In addition to the analysis of job advertisements, IS researchers utilized other methodologies such as surveys of IS managers and recruiters, focus groups of IS professionals, and interviews with IT managers to investigate industry requirements. The results gleaned from these other methodologies were quite different from the job analysis studies and indicated that so-called "soft skills" such as ability to communicate were more important in obtaining a job than technical knowledge (Litecky, 2012). To try to understand and explain the "hard-soft controversy" Litecky (2012) applied Image Theory, which explains that processes occur in steps. In the case of hiring, they posited there was a first filtration step to eliminate unacceptable candidates, followed by a second step for choosing the best candidate from the select group. Because soft skills are not easy to quantify they were not useful for the filtration step and were therefore, not included in job advertisements. However, the interview comprised the second selection step and because the hirer could determine whether the candidate communicated well, had leadership qualities, and the like, these soft skills then become the determinant as to whether the candidate was actually hired. In addition, during a face-to-face interview, the hirer might be more concerned with the applicant's long-term ability to contribute to the organization, which would include soft skills, rather than just the applicant's knowledge of a particular software package.

Longitudinal studies of job advertisements which covered from the 1970s into the 2010s (Gallivan 2004), (Harris (2012) shed additional light on this dichotomy. These studies included not just print advertisements but online advertisements, from sites such as www.dice.com and www.careerbuilder.com. Gallivan (2004) discovered that although technical skills continued to dominate print ads, online ads required a mix of both technical and non-technical skills (Gallivan, 2004). Litecky (2012) noted that online job boards and non-print media do not have the space limitations of print media and cost is not determined by the number of words used in the ad. This allows advertisers to list as many skills as they wish and the result is that soft skills are specified as well as hard skills. So perhaps, soft skills have been desired by employers all along but their specification was an added advertising cost which employers were not willing to expend.

Harris' 2012 study updated two earlier studies to include data from online job advertisements posted on www.careerbuilder.com. Data from the ads were parsed into tables and the context of each skill word checked to eliminate duplicates and to verify the word was, in fact, being used as a skill word. So for example, this method ensured "Access" was being used as the name of a software package and not as an everyday word. Using SQL queries the authors calculated the frequency of each skill and manually identified emergent skills that were brand-new. They found that there was a dramatic increase in the number of advertisements and the number of skills mentioned per advertisement. So, while there were 32 skills identified in the 1970s, this study identified 194 skills. On average, each ad mentioned seven skills, about double the number mentioned when only print media was used to advertise.

The ranking of various skills also changed over time. For example, Experience, which had been consistently ranked near the top, became the top skill in 2010 and appeared in over 90% of the ads; in 1970, it had only been in 17% of the ads. (Of course, there are now more applicants in the market who have experience as computer usage and training has exploded). Nevertheless, it represents one more hurdle particularly for the recent graduate. Remarkably, Communication which had not even been in the top ten became the second most frequently requested skill. To be clear, the communication skill requested is communication as it pertains to systems analysis and design, not a generalized reading or writing skill one might obtain through an English literature course or a course on public speaking. Bachelor was the third skill indicating degrees are more highly desired than they were in the dot-com era when it ranked number 14. Generalized demand for IS workers was high at that time and the number of educated workers was lower, so it may have been a supply and demand issue. Network and Database remained in the top ten skills although some specific database packages such as Oracle declined in frequency requested. Conversely, Java emerged as number 9, reflecting the general move to the Web. Other skills that declined markedly were: Web master, Unix, C/C++, and Visual Basic. Project Management and Security also moved up in the ranks, into the top 20, to number 11 and 12, respectively. Harris notes (p. 77) that the IS 2010 Model Curriculum addressed this increase in number and diversity of skills requested by employers, by recognizing the need for elective courses which enable the student to specialize (Topi, Valacich, Wright, Kaiser, Nunamaker, Sipior, and de Vreede 2010).

Litecky (2012) took online job advertisement analysis one step further by crawling five large US web sites including Monster.com, CareerBuilder.com, Dice.com, and SimplyHired.com, retrieving ads requiring degrees such as Computer Science (CS), Management Information Systems/Computer Information Systems (MIS/CIS), and Information Technology (IT) and reviewing more than 4,000 ads. Litecky only included skills which appeared in at least 5% of the ads and grouped the skills listed iinto three broad categories: Business Skills, Soft Skills, and General Technical Skills. An example of a Business Skill would be Contracting and Legal while an example of a Soft Skill would be Judgment & Decision Making.

The most frequent business skill requested was Managing/Supervision which appeared in 48% of the ads selected, followed by Administration/Quality in 29%. Financial, Project Management, Business Strategy, Accounting, and Marketing also appeared with frequencies above 10%. These results reflect the need for IS professionals to have general business skills and continues to differentiate the IS degree from computer science, for example. In the Soft Skills category, Leadership, Problem Solving, and Written Communication were the top three requested. In General Technical Skills, Security was requested most often with a frequency of 50%. Other top skills were: Testing, Certification, Programming, Office Applications, and Software Development. Litecky notes that although only 23% of the ads mentioned certification, for IS Security jobs, 50% did so. Also Litecky notes that now much of the IT work force comes from functional areas such as Accounting rather than an IT department, particularly due to growth in the adoption of enterprise system (ES) software. He speculates that integrating SAP into the MIS curriculum could increase enrollment (p. 41).

Hite (2012) queried a variety of online job bank web sites using key word lists from previous research studies. SQL was the skill in highest demand followed by JAVA. Other top ten skills requested included: Unix and Linux, HTML/DHTML, and C++. MSVisio and Visual Studio replaced earlier multimedia design software in popularity. Photoshop replaced earlier desktop publishing software, such as MS Publisher, and the entire category fell in popularity. With regard to enterprise software, SAP led, followed by Oracle and PeopleSoft. The author concluded that educators should emphasize general categories of IT instruction, such as database creation, rather than specific software and that some skills could most appropriately be taught in technical schools and community colleges.

Despite all these efforts to dovetail the IS curriculum to employer needs there remained a long-standing belief that the IS degree did not properly equip a student to perform an entry-level job successfully (Fang, 2005). Several studies revealed there was a gap between expected and actual skill performance (Cappel, 2001/2002). The recession of 2008 and the advent of IT outsourcing refocused research to redefine not just job skills for IS majors but entry-level skills in particular. This was in part due to declining enrollment in the IS major because students believed they would not be hired, despite the need for IS skills in the marketplace. (Burns et al, 2014).

Kennan (2008) narrowed the field of inquiry by only analyzing online job ads for positions suitable for early career graduates, that is graduates of IS programs with three years of experience or less. Kennan conducted a content analysis of 400 ads culled from the three major Australian online job sites, JobServe, seek, and MyCareer over a ten week period in 2006. Kennan downloaded the data into Simstat/Wordstat, a word counting software package which creates a hierarchical dictionary of terms. After consulting the literature, a team of knowledgeable academics and students manually grouped the terms into categories. Using Jaccard's coefficent of similarity measure and cluster analysis, they determined the key words which most commonly appeared together. The dominant cluster consisted of ten categories: Business and Systems Analysis; Management; Operations, Maintenance & Support; Communication Skills; Personal Characteristics; IS Development; Computer Languages; Data & Information Management; Internet; Intranet; Web Applications; and Software Packages. This cluster represents the core IS skills and job competencies required by employers for early stage graduates.

IS Development was the most frequently occurring term (78% of ads) and included skills such as Programming and Testing. Personal Characteristics and Communications Skills appeared in 75% of the ads with "ability to learn" appearing most frequently. These two categories were closely linked indicating that employers believe communication to be essential to development. Enterprise Resource Planning and Security and Project Management were least frequently mentioned and the authors surmise these skills would appear more often in ads targeted at graduates with more experience. Almost 50% of the ads also requested experience which could be an obstacle for recent graduates. The authors conclude that finding the right balance between business and technical skills remains a primary challenge for educators.

3. CURRENT MODEL IS CURRICULUM GUIDELINES

The current undergraduate IS program curriculum guidelines were developed in 2010 by a joint effort of the Association for Computing Machinery (ACM) and the Association for Information Systems (AIS) (Topi et al., 2010). These guidelines are referred to as the 2010 model. This 2010 model serves as a standard for numerous undergraduate IS and IS related programs in institutions around the world (Tehrani, 2015).

The 2010 model is summarized by Topi et al as follows:

"This revision has four broad key characteristics that have shaped the outcome significantly. First, the curriculum reaches beyond the schools of business and management. Previous versions of the IS curriculum have been targeted to a typical North American business school; this model curriculum is, however, guided by the belief that even though business will likely continue to be the primary domain for Information Systems, the discipline provides expertise that is critically important for an increasing number of domains. Second, the outcome expectations of the curriculum have been very carefully re-evaluated and articulated first in the form of high-level IS capabilities and then in three knowledge and skills categories: IS specific knowledge and skills, foundational knowledge and skills, and domain fundamentals. Third, the curriculum is structured so that it separates the core of the curriculum from electives with the intent of supporting the concept of career tracks. Finally, the design of this curriculum includes enough flexibility to allow its adoption in a variety of educational system contexts" (Topi et al, 2010).

Ultimately the 2010 model has seven core courses:

1. Foundations of Information Systems
2. Data and Information Management
3. Enterprise Architecture
4. IT Infrastructure
5. IS Project Management
6. Systems Analysis and Design
7. IS Strategy, Management, and Acquisition

In addition, the model includes several suggested electives. The authors of the model acknowledge that a complete collection of electives is not possible in a curriculum model but they do include some suggested sample electives. The suggested elective list includes Application Development, Business Process Management, Enterprise Systems, Human-Computer Interaction, IT Audit and Controls, Data mining / Business Intelligence, Collaborative Computing, Information Search & Retrieval, Knowledge management, Social Informatics, IT Security and Risk Management.

Appendix A is a matrix from the 2010 model that includes the core courses and sample electives mapped to a number of suggested career tracks. IS programs can use the matrix to tailor their core and elective course offerings to a specific job or job category and students can use the matrix to select a course sequence that builds a knowledge base for a specific job.

4. RESEARCH METHODOLOGY

This research was conducted using a "grounded theory" approach. Grounded theory was developed by the sociologists Barney Glaser and Anselm Strauss in the 1960's. In the grounded theory approach, conclusions are drawn and theories are produced by analyzing a body of data. In essence, the theories that are produced are "grounded" in the data (Glaser & Strauss, 1967).

For this project, job listings were collected and analyzed from several Internet web sites specializing in technology related employment. The listings were collected over a four month period from January to May 2017. In order to be included in the study, the job had to be technology based and entry level as indicated by the words "entry level" in the job listing or as indicated by requiring less than three years of experience. Also, the listing had to indicate that the job required an IS or IS related Bachelor's degree (Computer Information Systems, Information Technology Management, Business Information Systems, Management Information Systems, etc.).

For each job listing that met the criteria, a record was made of the various experience, knowledge, and skills required for the job. The knowledge and skills specified in the ads were categorized by type. The types included written, oral, and other types of communication skills, various technical skills, analytical skills, and business related skills. Also any other additional education or certifications required were recorded. The knowledge and skills for each category were then tabulated, summarized, and sorted in order of frequency of occurrence of specific words. In order to count word frequencies in each category a VBA macro published by Allen Wyatt (2016) was used. The following section shows the results of that analysis.

4. RESULTS

A total of 204 ads were examined in this study. Most of the ads were from Indeed (168), and the rest were from Monster, Dice, and Glassdoor. The ads ranged in dates from January 24 to May 11, 2017. The ads represented jobs in 36 states and the District of Columbia.

Years of Experience
Of all the ads examined, ninety-seven (97) ads specified a years-of-experience requirement, which typically ranged from 0 to 3 years.

Experience Skills
One hundred and fifty-eight ads (77%) required some experience. Overall, the most frequently words mentioned in conjunction with experience were support/supporting, technology, networks/networking, server, hardware/software, web, database(s), system(s), application(s), programming, troubleshooting, and helpdesk/help desk. The most frequently mentioned technical platforms were SQL, C/C++, Java, JavaScript, Linux and .Net, etc.

Communications Skills
One hundred thirty-two ads (65%) emphasized the requirement of excellent or strong written communications skills. One hundred thirty-seven ads (67%) required excellent or strong oral communications skills. Among those ads, one hundred thirty-one ads (64%) mentioned both oral and written communications skills. Separately, one hundred and four ads elaborated their communications requirements emphasizing the words or phrases of customer service, technical, interpersonal, team, and professional, etc.

Programming Skills

Table 1 Programming Skills

Word	Frequency
SQL	24
Java	18
Scripting	17
C	13
object-oriented	10
JavaScript	8
C++	8
.Net	7
Script	6
PowerShell	5
Shell	4
PYTHON	4
PHP	4
RUBY	3
OOP	3
PL	3
HTML	3
MySQL	3

Seventy-one ads (35%) required programming skills. The required programming languages are summarized in the following table. SQL, Java/JavaScript, C/C++ and .Net were most frequently mentioned.

Network Skills
Fifty-six ads (27%) specified a network skills requirement, with TCP/IP, LAN/WAN, DNS, protocols, firewall, server, and switches/routers being the most frequently mentioned.

Table 2 Network Skills

Word	Frequency
TCP/IP, IP	12
WAN	9
DNS	9
Protocols	8
LAN	8
Server	6
Firewalls	6
Switches	5
http	5
Routers	4
DHCP	4
Wireless	4

Database Skills

There are 31 ads (15%) that specified database skills with the most frequent words summarized in the following table.

Table 3 Database Skills

Word	Frequency
Relational	9
SQL	7
Oracle	5
SQL Server	4

Systems Analysis and Design Skills
There are 33 ads (16%) that specified systems analysis and design skills with the most frequent words summarized in the following table.

Table 4 Systems Analysis and Design Skills

Word	Frequency
SDLC	15
Design	7
Agile	6
project management	4
UML	4
Integration	2
Methodologies	2
Scrum	2
Waterfall	2
Iterative	2

Web Skills
There are 21 ads (10%) that specified Web skills with the most frequent words summarized in the following table. HTML, CSS, and .Net lead the table.

Table 5 Web Skills

Word	Frequency
HTML	9
Development	9
CSS	4
Design	4
Internet	2
.Net	2
Hosting	1
Security	1
ASP	1
JavaScript	1

Security Skills
Seventeen ads (8%) specified cyber security skills with the most frequent words summarized in the following table.

Table 6 Security Skills

Word	Frequency
Antivirus	3
Cyber	2
Firewalls	2
Virus	2

Other Technical Skills
Other skills requirements not fitting neatly into the above categories are summarized in table seven, which is based on content from 166 ads (81%) in this study. Microsoft Office (Excel, Word, PowerPoint) are the most common skills referred to in the ads. Windows, operating systems, server/servers, also seem to be a quite common requirement, followed by general technological categories like hardware, technology, applications, development, and PC/PC's, etc.

The bulk of the table represents a variety of technical skills that companies are looking for on the market. This skills category showcases the multifaceted nature of the IT field.

Table 7 Other Technical Skills

Word	Frequency
Microsoft/MS Office	128
Windows	60
operating system(s)	49
Hardware	37
server(s)	35
Technical	33
application(s)	22
IT	14
PC/PC’s	14
Technology	13
WMWare	13
Linux	13
Mac/Apple	12
Client	11
Virtualization	6
Mobile	6
Unix	5
SharePoint	4
Adobe	4
Infrastructure	4
Testing	4
Monitoring	4
Visio	4
R2	3
Mainframe	3
Microcomputer	3
ERP	3
Cloud	3
Android	3
PCS	3
iOS	3

Analytical and Business Skills

One hundred fifty-four ads (75%) specified analytical and business skills with the most frequent words summarized in the following table. Overall, they refer to a wide spectrum of soft skills such as problem solving, team work, analytical skills, time management, and self-motivation, among others.

Table 8 Analytical and Business Skills

Word	Frequency
Problem Solving	54
Team	42
Troubleshooting	42
Analytical	37
Detail	34
Independent	34
Time Management	29
Prioritize	25
Manage	25
Self Motivated	22
Business	19
Priorities	18
Supervision	14
Deadlines	13
Organized	9
Ethic	7
Analyze	7
Identify	7
Responsibilities	7
under pressure	6
Proactive	5
Logical	5
Driven	5

Certifications

Forty-one ads (20%) required or preferred certification. The most common certifications are summarized in the following table.

Table 9 Certifications

Word	Frequency
Network	14
Microsoft	13
CompTIA	7
Security	7
CCNA	5
Professional	4
Cisco	4

5. DISCUSSION

As previously stated this research has multiple goals. The first goal is to gain a general understanding of the current knowledge and skills that are most in demand by employers of the students graduating from IS programs. The second goal is to compare those skills to the most recent model IS curriculum that is used by many IS programs in colleges and universities around the world. Finally, the ultimate goal is to provide insight into a revised IS model curriculum that would prepare students to have the skills that are highest in demand by employers.

The results of this research show that the skills most in demand from employers are primarily soft skills and basic technology skills. Regarding soft skills, 75% of the prospective employers in the study were looking for employees with analytical and business skills such as problem solving and teamwork skills. Additionally, two thirds of the employers in the study were looking for employees with strong communication skills, both written and oral.

Eighty one percent of the ads were looking for general technology skills. Overall, Windows, server/servers, and Microsoft/MS Office (Excel, Word, PowerPoint) seem to be a quite common requirement, followed by other general technological categories like operating systems, technology, applications, development, and PC/PC's, etc.

Surprisingly, employers in the study were less likely to look for specific technical skills. About a third of the ads were looking for programming skills, 27% were looking for networking skills, and 15% were looking for database skills. Furthermore, only 16% were looking for systems analysis and design skills which is counterintuitive to the idea that the IS degree prepares students for the systems analyst job. Two skills that would seem to be in especially high demand in recent years, web development and cybersecurity, showed little demand in our study. Only 10% of the ads were looking for web development skills and 8% for cybersecurity.

When the skills required by employers in the study are compared to the suggested material covered in the 2010 model IS curriculum (shown in Appendix A) some interesting conclusions can be drawn. The first conclusion is that employers are primarily looking for soft skills but the model IS curriculum focuses on specific hard skills. Perhaps the argument can be made that the soft skills come from other sources. In particular soft skills may be learned through ancillary work in the core IS courses (such as group projects and presentations) or in the general education courses students are required to take.

Another interesting conclusion in the comparison of the employer required skills to the IS model curriculum involves the specific hard skills. This research shows that the hard skill most in demand by employers is programming. However, the IS model curriculum does not include programming as a core course. It does suggest that programming be included as an elective but that means that many students may choose not to take it. The model core curriculum does include network skills (as part of the "IT infrastructure" core course), database skills, and systems analysis and design skills, which are all skills that did show up in the study. Furthermore, most of the general technology skills most often sought by employers would be covered in the "Foundations of IS", "IT Infrastructure", and "Enterprise Architecture" core courses suggested by the model curriculum. The 2010 model curriculum purposely dropped the requirement for a personal productivity tools course (such as Excel or word processing) because the authors felt that most institutions required students to be proficient in these skills (Topi et al., 2010). However, our research shows that those skills are in high demand and subsequently it is important to ensure that students are getting those skills somewhere in the curriculum.

There are two categories of requirements that, according to the study, are in high demand by employers but are not explicitly or implicitly covered by the 2010 model IS curriculum. Those two areas are experience and external certifications. Twenty percent of the ads in the study were looking for applicants with an external certification and, as previously noted, 77% of the ads were looking for employees with experience.

The ultimate goal of this research is to suggest how the 2010 IS model curriculum should be amended to fulfill the requirements of employers in 2017. According to this research the following changes would be proposed. First, soft skills should be made more prominent in the curriculum. This could be accomplished through either adding a core course or adding soft skill coverage to existing core courses. Next, as programming is the highest demand technical skill, it may be a good idea to also make programming more prominent in the IS model curriculum. Again, this could be accomplished by adding a programming core course to the curriculum or by adding programming to one of the existing core courses. Finally, according to this research, the IS model curriculum should include an experiential component. This idea is already supported by many in the IS field, who feel that IS is an applied discipline and, as such, should emulate other applied fields such as medicine, engineering, and architecture by including an internship or other hands on experience in the curriculum (Moody and Buist 1999).

6. REFERENCES

Burns, T., Gao, Y., Sherman, C., Vengerov, A., & Klein, S. (2014). Investigating a 21st Century

Paradox: As the Demand for Technology Jobs Increases Why Are Fewer Students Majoring in Information Systems? *Information Systems Education Journal*, 12(4), 4-16.

Cappel, James J. (2002). Entry-level IS job skills: a survey of employers. *The Journal Of Computer Information Systems*, 42(2), 76-82.

Fang, X., Lee, S., & Koh, S. (2005). Transition of knowledge/skills requirement for entry-level IS professionals: an exploratory study based on recruiters' perception. *The Journal of Computer Information Systems*, 46(1), 58-82.

Gallivan, M.J., Turex, D. P. III, & Kvasny, L. (2004). Changing patterns in IT skill sets 1988-2003: a content analysis of classified advertising. *Database for advances in information systems. Summer*, 35(3), 64-87.

Glaser, B. G. & Strauss, A. L. (1967). The Discovery of Grounded Theory: Strategies for Qualitative Research, Aldine Publishing Company, Chicago, IL.

Harris, A H., Morris, S. A., Greer, T. H., & Clark, J. (2012). Information systems job market late 2970s-early 2010s. *The Journal of computer information systems,* 53(1), 72-79.

Hite, N. (2012). The status of IT skills in business during recessionary times: implications for educations. *Delta Pi Epsilon Journal*, LIV(2), 16-28.

Kennan, M. A., Cecez-Kecmanovic, D., Willard, P., & Wilson, C. S. (2008). IS Knowledge And Skills Sought By Employers: A Content Analysis Of Australian IS Early Career Online Job Advertisements. *The Australasian Journal of Information Systems,* 15(2), 1-22.

Knodel, M. (2016). Computer Opportunity And Curricula Based On Newspaper Advertisements. *The Journal of Data Education,* 22(4), 10-12.

Litecky, C. R., Arnett, K. P., & Prabhakar, B. (2004). The Paradox Of Soft Skills Versus Technical Skills In IS Hiring. *The Journal of computer information systems,* 45(1), 69-76.

Moody, D. & Buist, A. (1999). "Improving Links Between Information Systems Research and Practice – Lessons from the Medical Profession, *Proceedings of the 10th Australasian Conference on Information Systems*, 645-659.

Pierson, J.K, Kruck, S, & Teer, F, (2009). Trends In Names Of Undergraduate Computer Related Majors In AACSB-Accredited Schools Of Business In The USA, *Journal of Computer Information Systems,* 49(2), 26-33.

Tehrani, M. S. (2015). Towards A Consistency Of Information Systems Curriculum, *Issues in Information Systems*, 16(2).

Topi, H., Valacich, J., Wright, R., Kaiser, K., Nunamaker, Jr.,J.F., Sipior, J. & de Vreede, G.J. (2010. IS 2010 Curriculum Degree Programs Guidelines for Undergraduate in Information Systems: Association for Computing Machinery (ACM), Association for Information Systems (AIS), http://www.acm.org/education/curricula/IS %202010%20ACM%20final.pdf

Appendix A - Structure of the IS 2010 Model Curriculum (Topi et al, 2010).

Structure of the IS Model Curriculum:Information Systems specific courses

Career Track:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Core IS Courses:																	
Foundations of IS	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Enterprise Architecture	○	●	○	○	○	●	○	○	○	○	●	○	○	○	●	○	○
IS Strategy, Management and Acquisition	○	●	○	○	○	●	○	○	●	○	●	○	○	○	●	○	○
Data and Information Management	●	○	○	●	●	○	○	●	●	○	●	○	●	○	○	○	○
Systems Analysis & Design	●	●	●	○	○	○	●	○	○	○	○	○	○	○	●	●	●
IT Infrastructure	○	○	○	●	○	○	○	●	●	●	○	○	●	●	○	○	○
IT Project Management	●	○	○	○	○	●	○	○	○	○	●	○	○	○	●	●	●
Elective IS Courses:																	
Application Development	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●
Business Process Management		●	●			○	○	○		○	●				○		
Collaborative Computing						○								○			○
Data Mining / Business Intelligence		●		●	●	○	○	○	●		○	○	○	○	○		○
Enterprise Systems		●	●	○	○	○	●	●	○		●	●	○	○			
Human-Computer Interaction	●					○	○				○					●	
Information Search and Retrieval		○		○	●								○				●
IT Audit and Controls	○		●	○	○	○	○	●		●	○		○	○	○		○
IT Security and Risk Management	○			○	○	○	○	●	●	○	○		●	●	○		○
Knowledge Management		●		○		○	○			○							
Social Informatics													○		○		

A = Application Developer
B = Business Analyst
C = Business Process Analyst
D = Database Administrator
E = Database Analyst
F = e-Business Manager
G = ERP Specialist
H = Information Auditing and Compliance Specialist
I = IT Architect
J = IT Asset Manager
K = IT Consultant
L = IT Operations Manager
M = IT Security and Risk Manager
N = Network Administrator
O = Project Manager
P = User Interface Designer
Q = Web Content Manager

Key:

● = Significant Coverage

○ = Some Coverage

Blank Cell = Not Required

The Impact of Teaching Approaches and Ordering on IT Project Management: Active Learning vs. Lecturing

Christopher Sibona
sibonac@uncw.edu

Saba Pourreza
pourrezas@uncw.edu

Business Analytics, Information Systems, Supply Chain
University of North Carolina Wilmington
Wilmington NC 28403 USA

Abstract

This study explores the difference between both active learning and lecturing to teach Scrum project management in a university setting. The goal was to understand if one approach results in higher perceived learning over the other. Additionally, lesson ordering was examined to determine student preference of lecturing prior to or after an active learning exercise. Results suggest that students perceived they learned more from the active learning exercise compared to the lecture. Students preferred the active learning exercise compared to the lecture and found the active learning to be more engaging. Recommendations based on these findings are to use active learning exercises to teach Scrum project management in conjunction with lectures on the topic and to order the lessons with the lecture first followed by the activity.

Keywords: Scrum Project Management, Agile Project Management, Active Learning, Learning Styles, Pedagogy

1. INTRODUCTION

University faculty often desire to increase engagement of students in the classroom. One approach that has been successful is active learning. Active learning can engage students in the process of thoroughly learning a topic leading many educators to urge increased use of this approach (Bonwell and Eison, 1991). Furthermore, active learning exercises can engage students in higher-order thinking tasks such as analysis, synthesis and evaluation (Bonwell and Eison, 1991) which may help when approaching concepts that students may have little exposure to outside the classroom.

The traditional model of instruction, the lecture or transmission model, is more focused on remembering than internalization and deep understanding (Richardson, 1997). Alternatively, other approaches such as the constructivist approach to learning encourages students to engage, work, take ownership, and understand material by adding to known knowledge and building on new knowledge by exploring possibilities (Clark, 2008). This approach leads students to move beyond remembering material to more meaningful higher-order tasks. This study examines how an active learning exercise can engage students more thoroughly to understand project management in an information systems university setting.

The context for this study is Scrum project management (Schwaber and Beedle, 2002), where students learn the principles, roles, activities and iterations used to manage the system development life cycle. Scrum has been taught in both a lecture format and active learning activities. However, the challenge with teaching Scrum, especially to undergraduates, is many students have no prior work related experience and may have a difficult time understanding the topic. Thus, this study aims to explore how alternative approaches to teaching Scrum (i.e. active learning) may increase a student's perception of learning.

The students in this study were exposed to two approaches to learn Scrum project management concepts: lecture and active learning. This study examines student perceptions of these two approaches to the topic. Lecture is an example of aural learning style and activity lessons are example of kinesthetic learning style. Students were split into two groups some had the activity first followed by the lecture, while others had the lecture first followed by the activity. The goal was to examine differences in their perceptions of the lessons. The lessons cover a portfolio approach to the material as the content was covered through multiple methods (Lage et al., 2000). Student surveys were collected and analyzed to answer research questions about the lessons. The research questions examined in the current study include:

- What is the preferred approach for learning the Scrum, the active learning exercise or the lecture?
- Do students perceive they learn more during the activity or the lecture?
- What is the preferred order of the lessons - activity followed by lecture (AL) or lecture followed by activity (LA)?
- Is the activity or the lecture more engaging?

A post-hoc analysis was completed to determine how the preferred approach to learning was related to students' perceptions of their learning. We examined whether students who preferred one approach to learning (activity or lecture) perceived that they learned more in that approach, were indifferent to the approach, or whether they perceived they learned more in the other approach.

2. LITERATURE REVIEW

Active learning is a broad term for instructional methods that engage students through meaningful learning activities that require students to solve a problem or task (Prince, 2004; Bonwell and Eison, 1991). The task should be sufficiently complex that higher-order thinking is involved like analysis, synthesis and evaluation (Bonwell and Eison, 1991). Bonwell and Eison (1991, p. iii) define active learning as, "involving students in doing things and thinking about what they are doing."

Active learning can be further broken down based on the approach taken by the faculty and includes *collaborative learning, cooperative learning and problem-based learning* (Prince 2004). Collaborative learning is a group-based active learning technique where students work together in small groups to complete a common objective (Prince, 2004). A core element of collaborative learning is that students are working and interacting with each other instead of working individually. Cooperative learning is similar to collaborative learning where tasks are completed in small groups with the additional aspect that student progress is assessed at the individual-level (Prince, 2004). Students in cooperative learning settings will learn in a group, but are individually accountable for their learning outcomes. Problem-based learning is a technique where relevant problems are used to provide context and motivation for the learning objective (Prince, 2004). Problem-based learning often requires a student to apply their knowledge to solve a problem through self-directed learning (Prince, 2004).

A collaborative approach was used in this study for the active learning exercise where students worked together in small groups of three to four students to complete tasks using Scrum project management. The teams completed tasks from a sprint backlog following Scrum practices. Students were not assessed on their individual progress on the tasks but were to complete these activities to learn the process. The tasks were structured so that higher-order thinking would be involved and students could move beyond the mechanics to why the process works, and under what conditions Scrum is effective.

Teaching style and student learning styles work well when they are closely matched (Lage et al., 2000; Bishop and Verleger, 2013). When mismatches occur between teaching and student learning style a portfolio approach can be used (Lage et al., 2000). There are many different student centered learning styles and approaches to understanding learning style cited in the literature (Bishop and Verleger, 2013; Van Zwanenberg et al., 2000). Using multiple teaching styles in the classroom has been shown to increase student performance (Lage et al., 2000). Lujan and DiCarlo (2006) note that most first-year medical students preferred learning material through two or more presentation styles.

Fleming (2001) extends Eicher (1987) neuro-linguistic model into a sensory model known as VARK. Where VARK represents Visual (V), Aural

(A), Read/Write (R) and Kinesthetic (K). VARK is a learning style based on perceptual modes and instructional preference. This model is a preferred method for collecting, organizing, and interpreting information received (Hawk and Shah, 2007). In the VARK model, the visual learners prefer diagrams, charts, flow charts, graphs, different designs and pictures. Aural learners prefer lecture, topic discussions, group discussions, and seminar attendance. Read/ write learners prefer reading books and texts, handouts, articles, taking notes, and writing essays. Kinesthetic learners prefer real-life examples, physical activities, field trips, trial and error, constructing, working with models, laboratories, hands-on approaches, and collection of samples to understand problems and provide solutions for problems. Fleming's VARK questionnaire encourages learners to improve their learning by understanding their preferred modes of communication (Hawk and Shah, 2007). This study asked students to indicate their preferred method of learning and in which lesson they perceived they learned the most after both lessons were completed.

The research here followed a portfolio approach where students had both a lecture and an active learning session. This approach should allow more students to have the lessons presented in a manner that matches their learning style. A portfolio approach may also have the added benefit of increased student performance. According to Fleming (2006) any learning style that motivates learners to think about the way they learn, enhances learning since it is a step towards better understanding of the learning process. Fleming (2001) reports in his study when instructors match learning activities with students learning preferences, the students' performance improve in their courses since preferred learning modes bring flexibility for instructors and students so that both can change their behavior.

The research will also examine the relationship between the preferred lesson (presentation style) and student perceptions of learning. The lecture portion used an aural style and the activity used a kinesthetic style. The goal of using these multiple styles is to encourage deeper and higher-order learning.

3. RESEARCH METHOD

An online survey was used to collect the data for this study. Surveys from 155 students were collected over 2 semesters from five classes. Data was collected from three senior level information systems analysis classes and two introduction to management information systems classes. All sections were taught by the same instructor.

Study Design

The study was designed such that classes were randomly selected to one of two conditions: (1) activity first followed by the lecture (AL) or (2) lecture followed by the activity (LA).

The active learning exercise involved students folding origami using Scrum project management to complete the tasks. The folded origami represented software under development where students and instructor could measure progress of each task (user story). Students were provided packets of origami instructions and origami paper. Students formed groups of three to four students for the activity. Instruction about the Scrum process were provided and included a description of the product backlog (all the diagrams in the origami packet) a sprint backlog (a subset of the product backlog to be completed in a sprint), *day length* (for the purpose of the exercise, a day was 5 minutes), iteration length, scrum roles and daily questions. Students made estimates for the tasks in the sprint backlog, would hold a daily meeting and work through the *day* folding their origami*.* Progress would be measured after the iteration completed and adjustments could be made according to the Scrum process. A complete description of the exercise is beyond the scope of this paper.

Surveys were given to students after both the first and second lessons. Of the 155 students participating in the study, only 125 completed both surveys which resulted in dropping the 30 students who completed just one of the two surveys. The final participant count across the conditions was 41 students in the AL group and 84 students in LA group.

Survey questions were asked regarding the preferred method of delivery, perceptions of learning, and preferred order of delivery (AL or LA). All questions were scaled on a seven-point Likert-type scale. The survey did not use a forced choice design, i.e. a survey respondent could say they had no preference for method of delivery, perception of learning, or preferred order of delivery. A final question was asked to determine the level of engagement for the lesson (See Appendix A for a complete list of questions).

4. RESULTS

The analysis was performed using JMP Pro 13 from SAS. Results for each item were kept in their appropriate nominal or ordinal form throughout the analysis. For the first three questions, preferred method of delivery, perceptions of learning, and preferred order of lessons, the responses remain in nominal form (e.g. for preferred method of delivery the response could be lecture, exercise or no preference). Non-parametric testing was performed to determine if statistical difference were found for the research questions. Chi-square ($\chi2$) tests were performed and, where appropriate, Fisher's exact test was performed. Chi-square test and contingency tables were used to investigate student preferences and engagement. Fisher exact test was used to calculate more precise probabilities in situations where the sample size yields less than 5 expected values per cell.

Preferred Lesson

To answer the first research question, we asked students which lesson they preferred or if they liked them both about the same. Results suggest the activity was preferred by most students (69.6%), followed by those who had no preference (20.8%) then those who preferred the lecture (9.6%). $\chi2$ probability results were <.0001 indicating that the null hypothesis is not supported. Thus, the results suggest that the answer to the question concerning preferred approach is that students preferred the activity more than the lecture (see Figure 1).

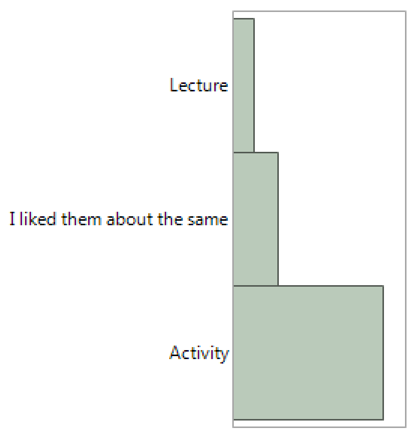

Figure 3: Preferred Lesson

Perceived Learning

Our next research question attempts to answer the question of perceived learning based on the approach. Results suggest the students perceived they learned the most during the activity compared to lecture. 45.6% of students perceived they learned most in the activity, 31.2% perceived they learned about the same in both the activity and lecture, and 23.2% perceived they learned more in the lecture. $\chi2$ probability results were <0.0080 indicating that the null hypothesis is not supported (see Figure 2).

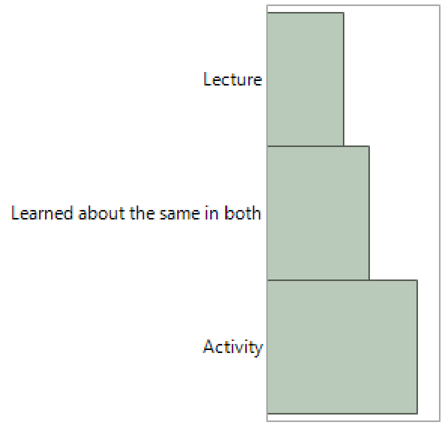

Figure 4: Perceived Learning

Preferred Order of Lessons

The next question is whether students have a preferred lesson ordering. Students preferred the order of the lessons to be lecture followed by the activity. 50.4% preferred lecture followed by the activity. 32.8% preferred the activity followed by the lecture. 16.8% showed no preference. $\chi2$ probability results were <0.0001 indicating that the null hypothesis is not supported (see Figure 3).

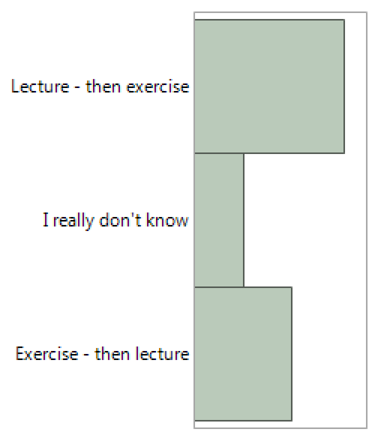

Figure 5: Preferred Order of Lessons

Furthermore, to ensure there was not difference based on ordering effects, we examine the difference between those who received the lecture first to those participating in the activity first. Results show that the preferred lesson (lecture/activity/no preference) did not vary (in a statistically significant way) based on the order that students experienced the lessons. The preferred lesson was the activity for both groups. The activity was favored by 69.6% of all students. Students who had the lecture first liked the activity at a higher rate than those who had the activity first (73.8% vs. 61.0%), but the difference is not statistically significant. χ2 probability results were <0.1200 indicating that the null hypothesis is supported. Fisher's Exact Test probability was 0.1239 and also indicates no statistically significant difference are present (see Figure 4).

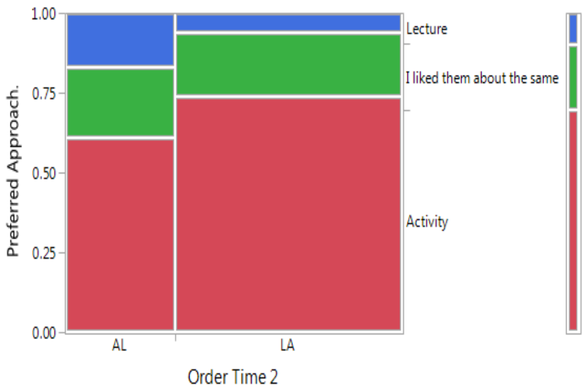

Figure 6: Preferred Lesson by Order

Perceptions of Learning by Order
A similar analysis was conducted concerning learning perceptions to answer whether those who had the activity→lecture condition (AL) perceived they learned more compared to those who had the lecture→activity condition (LA). There was no statistically significant difference between learning perceptions across the two conditions. In other words, those who had the AL condition did not perceive they learned more (or less) than those who had the LA condition

Students in both groups said they learned the most in the activity. The activity was selected by 45.6% of students as the lesson by which they learned the most. In fact, those who had the lecture first (LA) perceived they learned the most through the activity compared to those who had the activity first (AL) (51.2% vs. 34.1%). However, the difference is not statistically significant as the χ2 probability results were <0.1929 indicating that the null hypothesis is supported. Fisher's Exact Test was 0.1874 and indicates no statistically significant difference (see Figure 5).

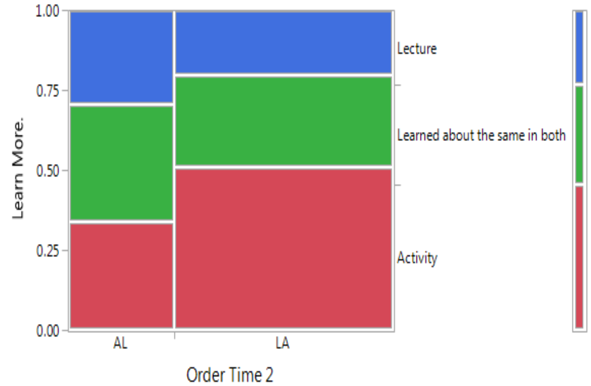

Figure 7: Perceptions of Learning by Order

Preferred Order of Lessons by Order
The results in this section determine whether students who experienced one order of instruction indicated that they would prefer the same order of instruction or indicated that they would prefer to have experienced a different order. For example, did students who experienced the activity followed by the lecture have a preferred order of lessons that is different from students who had the lecture followed by the activity?

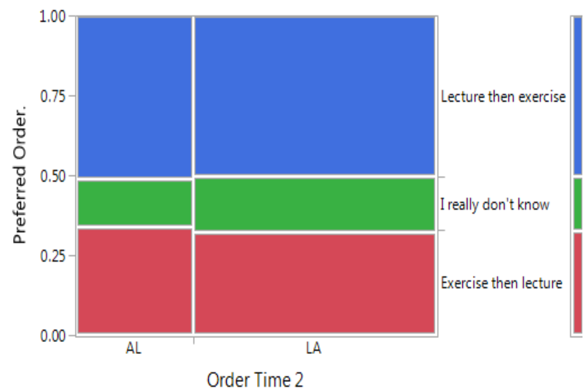

Figure 8: Preferred Order of Lessons by Order

Students who had the activity→lecture did not prefer a lesson ordering different from students who had the lecture→activity in a statistically significant way. A majority of students (50.4%) said that the preferred order of lessons was to have the lecture then the activity. Students who had the lecture first perceived that the best order was lecture then activity at a lower rate than students who had the activity first (50.0% vs. 51.2%), but the difference is not statistically significant. χ2 probability results were <0.8976 indicating that the null hypothesis is supported. Fisher's Exact Test was 0.9341 and also indicates no statistically significant differences. See Figure 6.

Lesson Engagement

The last research question focuses on what was more engaging, the activity or lecture. To examine this, the results are broken down across the times they received each lesson. Recall that a post lesson survey was conducted after each lesson at time 1 and time 2. Thus, the results below will first discuss engagement after the first lesson followed by a discussion of engagement following the second lesson.

The results across both conditions (AL and LA) after lesson 1 suggest that those receiving the activity first found it to be very engaging (50% of students strongly agreed that the activity was engaging). This is compared to those receiving the lecture first where only 7.2% of students strongly agreed that the lecture was engaging. These results were found to be statistically significant suggesting the active learning exercise to be much more engaging. Students in the activity group strongly agreed that the activity was engaging. χ2 probability results were <0.0001 indicating that the null hypothesis is not supported. Fisher's Exact Test probability was < 0.0001 and also indicates statistically significant differences are present (see Figure 7).

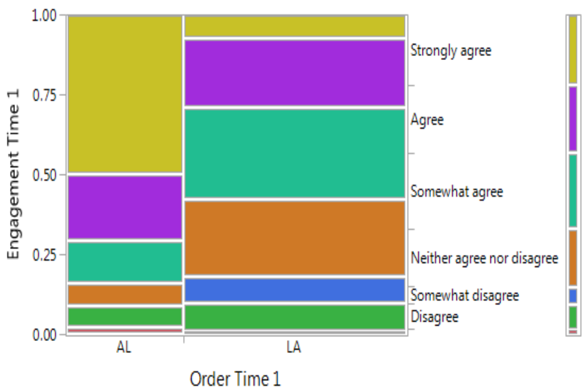

Figure 7: Engagement First Session by Order

The lesson 2 results found that 47.7% of students who had the activity in the second class strongly agreed that the activity was engaging whereas only 18.6% of students who had the lecture in the second class strongly agreed that the lecture was engaging. Again, this was statistically significant supporting the prior lesson 1 results that student engagement was strongest for activity based learning compared to lecture. Students in the activity group strongly agreed that the activity was engaging. χ2 probability results were <0.0003 indicating that the null hypothesis is not supported. Fisher's Exact Test probability was < 0.0001 and indicates statistically significant differences are present (see Figure 8).

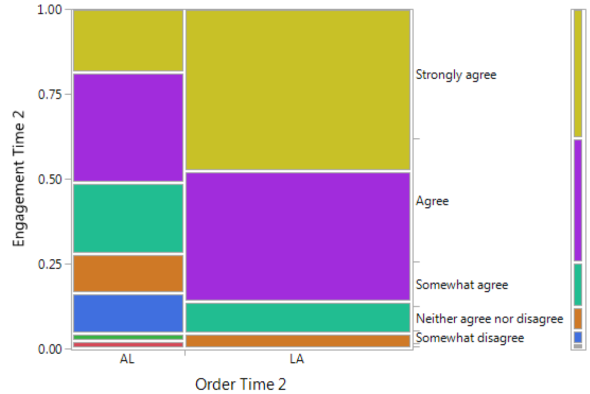

Figure 8: Engagement Second Session by Order

Preferred Lesson and Perceptions of Learning Relationship

Finally, a post-hoc analysis was conducted to understand the relationship between preferred lesson (activity/lecture/no preference) and perceptions of learning (learned more in lecture/activity/learned about the same in both approaches). The analysis found that students perceived that they learned the most in the activity (45.6%) compared to the lecture (23.2%) and compared to those who said they learned about the same amount in each lesson (31.2%). Thus, there is a statistically significant difference between the preferred lesson and where the student perceived they learned the most. Students who preferred the activity (69.6% of learners) perceived that they learned the most in the activity (62.1%), whereas 23.0% felt they learned the same in both approaches and 14.9% felt they learned more in the lecture.

A minority of students (9.6%) preferred the lecture to the activity or had no preference, but of these students 58.3% of them felt they learned more in the lecture. Students who preferred the lecture chose the activity as the lesson where they learned more at lower levels (14.9%). Students who said they had no preference in lesson (20.8%) said they learned about the same amount in both lessons at a higher level (57.7%). χ2 probability results were <0.0001 indicating that the null hypothesis is not supported. Fisher's Exact Test probability was < 0.0001 and also indicates statistically significant difference are present (see Figure 9).

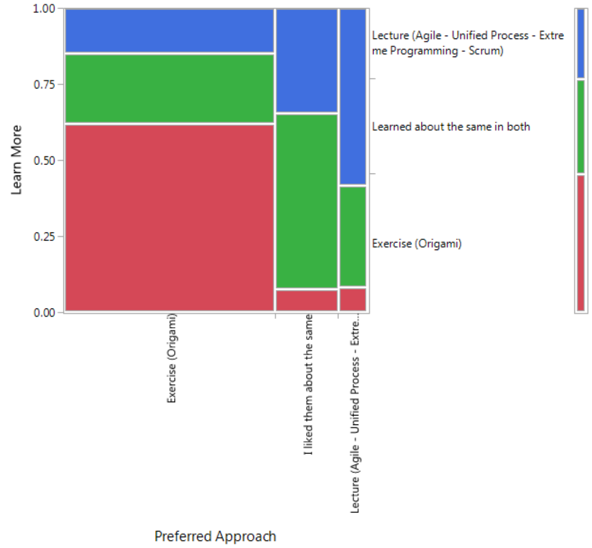

Figure 9: Preferred Lesson and Learning Relationship

5. DISCUSSION

The purpose of this study is to determine preferred method of instruction, ordering of the lessons, and perceptions of learning.

A clear majority of students (69.6%) preferred the activity lesson for Scrum project management. A small minority of students (9.6%) preferred the lecture to the activity and approximately one fifth of students (20.8%) were indifferent to the method - they liked the activity and lecture at about the same level. These results are consistent with Bonwell and Eison (1991), where more students prefer active learning to lecture methods.

Students perceived that they learned the most in the activity (45.6%), and about a quarter (23.2%) perceived they learned more in the lecture. Approximately 1/3 of students (31.2%) were indifferent to where they learned the most - they perceived that they learned about the same amount in the lecture and the activity. These results indicate that a portfolio approach may be used to match student learning styles with an appropriate method. Every learner prefers one or combination of different perceptual modes from the VARK learning style (Hawk and Shah, 2007). When instructors bring different learning styles such as, kinesthetic and aural described in VARK more students are able to learn more effectively (Alkhasawneh, Mrayyan, Docherty, Alshram, Yousef, 2008). The lessons on project management included a lecture component that focused on aural learning and an activity that focused on kinesthetic learning.

Aside from preferred method of instructions, the results of the study also suggest that ordering may play a role in student learning. Approximately half of the students (50.4%) would prefer the lessons to be ordered with the lecture first followed by the activity. This is compared to less than one third of students (32.8%) preferring the activity first and 16.8% of students were indifferent. Providing the lecture first would fit the flipped classroom style of instruction where students learn about the concepts first and then delve deeper into the material in the classroom. There are advantages and disadvantages to either approach. Students were less engaged in the lecture portion than the activity (see Figures 7 & 8) so they are more passive about the instruction. Student who had the lecture→activity may not realize the importance of the material because they are less engaged. It is somewhat easier to engage the students who had the activity→lecture during the lecture because they were more engaged, overall, in the lesson. The lecture becomes more cogent to the learner when the student experienced the activity first. A potential downside to the activity→lecture is that some amount of preparation must be completed before the students can start the activity so that the lesson can be learned. The game mechanics must be easy enough to learn and complex enough that the activity reflects a real-life situation and still have the student learn from the experience (Baker et al., 2005).

Several analyses were performed to determine whether the ordering of the lessons (AL or LA) had statistically significant differences on the preferred approach, students' perceptions of learning, and the preferred order of the lessons. Students who had the lecture→activity preferred the activity at higher levels (73.8%) compared to those who had the activity→lecture (61.0%), but the differences were not statistically significant. In general, we can say that the experience of having activity→lecture or lecture→activity did not change a student's preference for the activity over the lecture. The number of students who expressed no preference for both groups is similar - approximately 20% of students liked both approaches about the same. The lecture was the preferred approach by larger percentages of students who were in the activity→lecture cohort (17.1%) compared to the lecture→activity cohort (6.0%) but this difference was not statistically significant. This may be similar to the trade-offs discussed above, the lecture may be more cogent to those who experienced the activity first, but not so much so that the lesson order preference changed from activity to lecture or vice-versa in a statistically significant manner.

Analysis on whether the lesson order (AL or LA) affected the student perceptions of learning was not statistically significant. The distributions in terms of where students thought they learned the most by lesson was more evenly distributed. Overall, students indicated that they learned the most in the activity (45.6%), followed by stating they learned about the same in both (31.2%), then the lecture (23.2%). Those who had the lecture→activity thought they learned more in the activity (51.2%) compared to the activity→lecture group (34.2%) but this difference is not statistically significant. Those who had the activity→lecture had higher levels stating that they learned more in the lecture (29.3%) compared to the lecture→activity group (20.2%) but this difference is not statistically significant. The distributions for the activity→lecture group were much more evenly distributed regarding their perceptions of where they learned the most. In the activity→lecture group, 36.5% said they learned about the same in both lessons, 34.2% said they learned more in the activity and 29.3% said they learned most in the lecture. The distributions for the lecture→activity group were much less evenly distributed regarding their perceptions of where they learned the most, the majority (51.2%) said they learned the most in the activity. The remaining members in the lecture→activity group, 28.6% said they learned about the same in both lessons and 20.2% said they learned most in the lecture. Overall, it appears that the activity was where students felt they learned the most and that the lesson order did not have a statistically significant impact on where students perceived the learned the most.

Analysis showed that actual lesson ordering (AL or LA) on the preferred lesson order (activity→lecture or lecture→activity) had little impact how the students preferred lesson order. That is, a majority of the students (50.4%) felt that the content should be delivered with the lecture first then the activity. Only small differences exist between those who experienced activity→lecture or lecture→activity in the classroom.

The post-hoc analysis regarding the students who experienced different lesson orders revealed only small differences in their perceptions in terms of their preferred ordering (activity or lecture), where students felt they learned the most content (activity or lecture), and lesson order (lecture followed by activity or activity followed by lecture). Students generally preferred the activity, felt they learned the most in the activity, and preferred the lesson order to be lecture followed by activity.

The analysis found statistically significant differences in the relationship between the preferred lesson (activity/lecture/no preference) and student perceptions of where they learned the most (activity, lecture or learned about the same in both approaches). Students who preferred the activity felt they learned the most in the activity, students who were indifferent to the approach felt that they learned about the same in both lessons, and those who preferred the lecture felt they learned the most in the lecture. In terms of class size, it is important to note that most students (69.6%) preferred the activity compared to the smaller group who preferred the lecture (9.6%), and 20.8% of students who were indifferent.

Students who preferred the activity perceived they learned the most in activity (62.1%). There were students who preferred the activity and said that they learned about the same in both lessons (23.0%), and students who said they learned the most in the lecture (14.9%). Students who had no preference for activity or lecture (20.8%) felt they learned about the same in both approaches (57.7%), then the lecture (34.6%), then the activity (7.7%). And students who preferred the lecture (9.6%) felt they learned the most in the lecture (58.3%). There were students who preferred the lecture and said that they learned about the same in both approaches (33.3%), and students who said the learned the most in the activity (8.4%). Students who preferred one lesson type and then stated that they learned more in the other lesson type were a minority of students. Generally speaking, if students preferred one lesson type they said they learned more in that lesson type. Using activities in the classroom will likely increase student perceptions that they are learning more in the classroom, perhaps because they are more engaged with the material. A small minority of students of approximately 10% preferred the lecture to the activity. Students who preferred the lecture have a 58.3% probability that they learned more in the lecture, where students who preferred the activity had a 62.1% probability that they learned the most in the activity. Students who were indifferent to the approach have a 57.7% probability that they learned about the same in both approaches. These results support that diversity in teaching styles may increase student performance (Lage et al., 2000; Lujan and DiCarlo, 2006).

Analysis on engagement with the activity and lecture show that students were more engaged with the activity at statistically significant levels. Levels of engagement were collected after each class session. 50% of students who had the activity in the first class strongly agreed that the activity was engaging. 7.2% of students who had the lecture in the first class strongly agreed that the lecture was engaging. There are statistically significant differences in how the lesson (activity or lecture) engaged the students where students were more likely to strongly agree that the activity was engaging. Similar differences exist for the second class session. 47.7% of students who had the activity in the second class strongly agreed that the activity was engaging. 18.6% of students who had the lecture in the second class strongly agreed that the activity was engaging. There are statistically significant differences in how the lesson (activity or lecture) engaged the students where students were more likely to strongly agree that the activity was engaging. Higher levels of engagement are expected in active learning environments and these findings are consistent with the expectations (Prince, 2004; Bonwell and Eison, 1991).

6. LIMITATIONS

There are many active learning methods like co-operative learning, problem-based learning, flipping (inverting) the classroom, inquiry-based learning, guided classroom discussion, etc. that were not investigated in this research. It would be difficult to draw definitive conclusions beyond teaching Scrum project management as an active learning exercise. Project management approaches may be particularly well-suited to active learning, as evidenced by the more than 150 agile games available online (see TastyCupcakes.org for additional examples). Additionally, the results are consistent with past research on active learning and constructivist approaches where student engagement is increased through these approaches (Bonwell and Eison, 1991; Richardson, 1997).

There may be ordering effects which may limit the generalizability of the results. The approach taken in this research did not attempt to have students who were part of the activity→lecture group gain exposure to the concepts prior to the active learning exercise. Potentially, flipping the classroom for the activity→lecture students may diminish or eliminate the need to have the lecture in a classroom setting. Flipping the classroom may have prepared the students better for the activity that students completed. The activity was designed to be quick to understand but more familiarity with the content may have been helpful. However, our results suggest that which group the students were assigned to did not have a significant impact on their perceptions.

The instructor who taught both the activity and lecture may be better at facilitating active learning exercises then lectures. It was not the goal of the instructor to have a low engagement lecture but other instructors may be better at this approach. Students have different learning styles with which they are comfortable and teachers have different aptitudes with different teaching styles (Lage et al., 2000).

Another potential limitation may be comprehensiveness of the activity. The lesson covering Scrum project management was not meant to be comprehensive. Rather, it was designed as a primer on the subject, thus the lesson did not cover everything required to effectively use the method. The lesson covers basics of roles, activities and processes for team members. The product owner task was controlled by the instructor and did not attempt to cover all of the decision making for product owners in Scrum.

Finally, this research measures student perceptions of learning, not actual learning. All questions are a self-assessment in which students provide their perceptions of learning about the activity and lecture lessons. The students' assessment about their learning may be more tied to their preferred approach then their actual learning. This limitation can be addressed by including objective measures of learning outcomes in the future. This research does not investigate a causal relationship between the preferred lesson and perceptions of learning.

7. CONCLUSION

This study examines multiple aspects of active learning and lecture approaches in a university setting. The context for this study is instruction on Scrum project management (Schwaber and Beedle, 2002) to undergraduate students. Most students preferred the active learning exercise compared to the lecture. The results are consistent with the literature (Bonwell and Eison, 1991; Lage et al., 2000) where students tend to prefer active learning to lectures.

Most students perceived that they learned more in the activity compared to the lecture. The results show a diversity in student learning styles; while most students preferred the activity, a minority of students preferred lecture and these students felt that they learned more in the lecture compared to the activity. The results support using a portfolio approach to teaching the material where multiple methods are used to cover the material. Students preferred the content to be ordered with the lecture first followed by the activity rather than having the activity first followed by the lecture. This approach would be similar to approaches by instructors advocating a flipped classroom approach (Lage et al., 2000; Bishop and Verleger, 2013). Students found the activity to be a more engaging activity compared to the lecture as expected in an active learning approach (Bonwell and Eison, 1991; Prince, 2004). Recommendations based on this research would be to include active learning exercises to teach project management approaches, deliver the content with a lecture first followed by the activity, and continue to teach with a lecture session and an active learning session. Students likely moved beyond the basics of remembering information to higher-order thinking like analysis and evaluation by delivering these lessons through both active learning and lecture formats.

8. REFERENCES

Alkhasawneh, Israa M., et al. "Problem-based learning (PBL): assessing students' learning preferences using VARK." Nurse Education Today 28.5 (2008): 572-579.

Baker, A., E. O. Navarro, and A. van der Hoek (2005). An experimental card game for teaching software engineering processes. *Journal of Systems and Software 75*(1-2), 3–16. Software Engineering Education and Training.

Bishop, J. L. and M. A. Verleger (2013). The flipped classroom: A survey of the research. In *ASEE National Conference Proceedings, Atlanta, GA*, Volume 30, pp. 1–18.

Bonwell, C. C. and J. A. Eison (1991). *Active Learning: Creating Excitement in the Classroom. 1991 ASHE-ERIC Higher Education Reports.* ERIC.

Clark, J. (2008). Powerpoint and pedagogy: Maintaining student interest in university lectures. *College teaching 56*(1), 39–44.

Fleming, Neil. "Learning Styles Again: VARKing up the right tree!." Educational Developments 7.4 (2006): 4

Fleming, Neil D. Teaching and learning styles: VARK strategies. IGI Global, 2001.

Hawk, Thomas F., and Amit J. Shah. "Using learning style instruments to enhance student learning." Decision Sciences Journal of Innovative Education 5.1 (2007): 1-19.

Freeman, S., S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences 111*(23), 8410–8415.

Lage, M. J., G. J. Platt, and M. Treglia (2000, Winter). Inverting the classroom: A gateway to creating an inclusive learning environment. Journal of Economic Education 31(1), 30. Copyright HELDREF PUBLICATIONS Winter 2000.

Lujan, H. L. and S. E. DiCarlo (2006). First-year medical students prefer multiple learning styles. Advances in Physiology Education 30(1), 13–16.

Prince, M. (2004). Does active learning work? a review of the research. Journal of engineering education 93(3), 223–231.

Richardson, V. (1997). Constructivist teaching and teacher education: Theory and practice. Constructivist teacher education: Building a world of new understandings, 3–14.

Schwaber, K. and M. Beedle (2002). Agile software development with Scrum, Volume 1. Prentice Hall Upper Saddle River.

Van Zwanenberg, N., L. Wilkinson, and A. Anderson (2000). Felder and Silverman's index of learning styles and Honey and Mumford's learning styles questionnaire: how do they compare and do they predict academic performance? Educational Psychology 20(3), 365–380.

A Appendix: Related Survey Questions

1. Did you prefer the lecture or exercise more?
   (a) Exercise (Origami)
   (b) Lecture (Agile - Unified Process - Extreme Programming - Scrum)
   (c) I liked them about the same
2. Where did you learn more?
   (a) Exercise (Origami)
   (b) Lecture (Agile - Unified Process - Extreme Programming - Scrum)
   (c) Learned about the same in both
3. What order would you prefer the classes to be in?
   (a) Exercise - then lecture
   (b) Lecture - then exercise
   (c) I really don't know
4. This exercise/lecture was engaging
   (a) Strongly disagree
   (b) Disagree
   (c) Somewhat disagree
   (d) Neither agree or disagree
   (e) Somewhat agree
   (f) Agree
   (g) Strongly Agree

B Perceptions of Learning by Preferred Approach

The figures show the relationship between students who had a stated preference for the delivery of the content (prefer activity, prefer lecture, and no preference) and where the student perceived they learned the most. See Figures 10, 11 and 12. Students who preferred the activity indicated they learned the most in the activity (Figure 10). Students who had no preference for the approach indicated the learned about the same in both approaches (Figure 11). Students who preferred the lecture indicated they learned the most in the lecture (Figure 12).

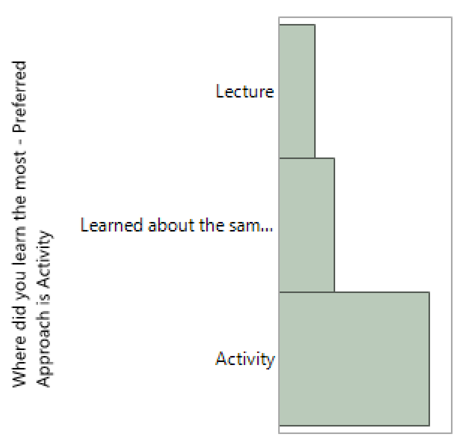

Figure 10: Perceptions of Learning Students who Preferred the Activity

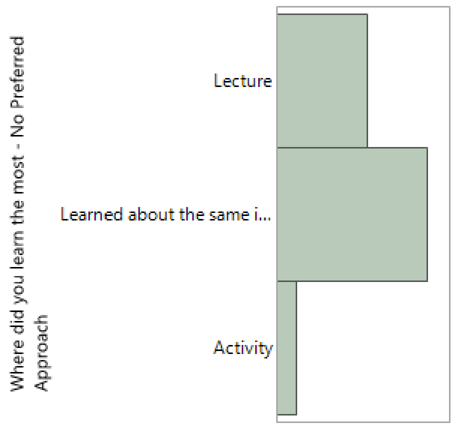

Figure 11: Perceptions of Learning by Students who had No Preference

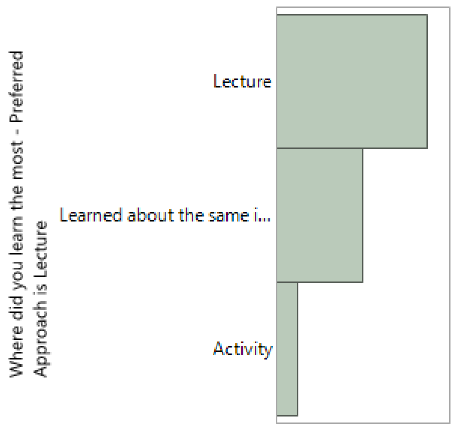

Figure 12: Perceptions of Learning by Students who Preferred the Lecture